### Frontmatter

**Editors**

SIGAPP FY'16 Quarterly Report  
J. Hong  

SAC 2016 Report  
H. Haddad

### Selected Research Articles

- **Regex-based Entity Extraction with Active Learning and Genetic Programming**  
  A. Bartoli, A. Lorenzo, E. Medvet, and F. Tarlao  
  Page: 7

- **Playing with Sybil**  
  F. Saab, A. Kayssi, I. Elhajj, and A. Chehab  
  Page: 16

- **A Retrospective Analysis of SAC Requirements Engineering Track**  
  J. Vilela, E. Gonçalves, A. Holanda, J. Castro, and B. Figueiredo  
  Page: 26

- **Efficient SQL Querying on Embedded Devices using Pre-Compilation**  
  G. Douglas and R. Lawrence  
  Page: 42
SIGAPP FY’16 Quarterly Report

April 2016 – June 2016
Jiman Hong

Mission

To further the interests of the computing professionals engaged in the development of new computing applications and to transfer the capabilities of computing technology to new problem domains.

Officers

<table>
<thead>
<tr>
<th>Role</th>
<th>Name</th>
<th>Institution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chair</td>
<td>Jiman Hong</td>
<td>Soongsil University, South Korea</td>
</tr>
<tr>
<td>Vice Chair</td>
<td>Tei-Wei Kuo</td>
<td>National Taiwan University, Taiwan</td>
</tr>
<tr>
<td>Secretary</td>
<td>Maria Lencastre</td>
<td>University of Pernambuco, Brazil</td>
</tr>
<tr>
<td>Treasurer</td>
<td>John Kim</td>
<td>Utica College, USA</td>
</tr>
<tr>
<td>Webmaster</td>
<td>Hisham Haddad</td>
<td>Kennesaw State University, USA</td>
</tr>
<tr>
<td>Program Coordinator</td>
<td>Irene Frawley</td>
<td>ACM HQ, USA</td>
</tr>
</tbody>
</table>

Notice to Contributing Authors

By submitting your article for distribution in this Special Interest Group publication, you hereby grant to ACM the following non-exclusive, perpetual, worldwide rights:

- to publish in print on condition of acceptance by the editor
- to digitize and post your article in the electronic version of this publication
- to include the article in the ACM Digital Library and in any Digital Library related services
- to allow users to make a personal copy of the article for noncommercial, educational or research purposes

However, as a contributing author, you retain copyright to your article and ACM will refer requests for republication directly to you.

SIGAPP Award Winners

Dr. Sung Shin, professor at South Dakota State University, USA, received the 2016 SIGAPP Outstanding Service Award in recognition of his outstanding service to the SIGAPP, and Dr. Dongwan Shin, director of Secure Computing Laboratory at New Mexico Tech, USA, received the 2016 SIGAPP Distinguished Service Award for his exceptional service to the annual SAC conference.
The 31<sup>st</sup> Annual meeting of the ACM Symposium on Applied Computing (SAC) was held in Pisa, Italy, April 4 to 8, 2016. The conference was hosted by the University of Pisa and Scuola Superiore Sant’Anna University. The conference attracted over 480 attendees including local host committee and graduate students from the host Institution. The event was a great success, with high attendance rate, due to the exceptional dedication of the local organizing committee and the host Universities. The Steering and Organizing Committees extend their thanks and gratitude to the local hosts for their support. This year, the conference featured technical sessions, tutorials, keynote sessions, posters, and Student Research Competition program.

The Call for Papers attracted 1047 submissions from 58 countries. All submitted papers underwent the blind review process and 252 papers were finally accepted as full papers for inclusion in the conference proceedings and presentation during the Symposium. The final acceptance rate for SAC 2016 is 24% among all tracks. In addition, 111 papers that received high review scores were invited as short papers for presentation during the Posters Program. The four-day Technical Program consisted of two keynote sessions and research presentations from all 37 tracks covering a wide range of topics on applied computing and emerging technologies. The Call for Track Proposals resulted in accepting 37 tracks. The selections were made based on the success of those Tracks in the previous editions of SAC as well as targeting new and emerging areas in applied computing. The Tracks were organized into five different themes: AI & Agents, Distributed Systems, Information Systems, Software Design & Development, and System Software & Security. The conference proceedings and the technical presentations were focused around these themes to form a series of related track sessions. For more details please visit [http://www.sigapp.org/sac/sac2016/](http://www.sigapp.org/sac/sac2016/).

The Call for Student Research Abstracts, for participation in the Student Research Competition (SRC) Program, attracted 47 submissions, of which 22 submissions were invited for participation in the program. The SRC program is sponsored by Microsoft Research. Invited students participated in poster display and oral presentations. A committee of five judges evaluated the posters and selected five winners for the second round (oral presentations). The judges then selected top three winners from the oral presentations round. The winners were recognized during SAC Banquet and presented with award medals and cash prizes. The winners of SAC 2016 SRC Program are below, and the first place winner will advance to the SRC Grand Finals.

- **First Place:** A 3D-Cellular Automata based Pseudo-random Number Generator. Rosemary Koikara, Kyungpook National University, South Korea
- **Second Place:** Internet of Things-Based Smart Classroom Environment. Amir Atabekov, Kennesaw State University, USA
- **Third Place:** An Efficient and Effective Link-based Similarity Measure in Social and Information Network. Masoud Reyhani Hamedani, Hanyang University, South Korea

In the Technical Program, a best paper was selected from each Theme. A committee led by the Program Chairs evaluated nominated papers. The best paper winners are:

- **AI and Agents Theme:** Discovering Discriminative Graph Patterns from Gene Expression Data. Fabio Fassetti, Simona E. Rombo, and Cristina Serrao. University of Calabria, Italy and University of Palermo, Italy
- **Distributed Systems Theme:** Rich Cloud-based Web Applications with CloudBrowser 2.0. Xiaozhong Pan and Godmar Back, Virginia Tech, USA
**Information Systems Theme:** Improving SQL Query Performance on Embedded Devices using Pre-compilation. *Graeme Douglas, and Ramon Lawrence, University of British Columbia, Canada*

**System SW & Security Theme:** Optimizing Record Data Structures in Racket. *Tobias Pape, Vasily Kirilichev and Robert Hirschfeld, University of Potsdam, Germany*

In the Posters Program, a committee led by the Posters Chair evaluated all participating posters and selected best poster. The best poster winner is:

Towards Spreadsheet Integration using Entity Identification Driven by a Spatial-Temporal Model. *Ramoza Ahsan, Rodica Neamtu, and Elke Rundensteiner, Worcester Polytechnic Institute, USA*

The *Tutorials* program featured 6 invited tutorial presentations. The tutorials covered a variety of topics and attracted over 120 attendees. The success of SAC 2016 was made possible through the hard work of many people from the scientific community who had volunteered and committed many hours to make it a successful event, especially, the Track Chairs and their Program Committees. On behalf of the Organizing and Steering Committees, we congratulate all of the authors for their successful work. We also wish to thank all of those who made this year's technical program a successful one, including the speakers, track chairs, reviewers, program committee members, session chairs, presenters, and attendees.

The preparation for SAC 2017 is underway. The conference will be held in Marrakech, Morocco (April 3 - 7, 2017). The conference is hosted by the University of Quebec at Montreal, Canada; University Cadi Ayyad (UCA) of Marrakech, Morocco; Mohamed V University of Rabat - Mohammadia School Of Engineers (EMI), Rabat Morocco; and National School of Applied Sciences (ENSA) of Kenitra, Morocco. We welcome your submission to SAC 2017 ([http://www.sigapp.org/sac/sac2017/](http://www.sigapp.org/sac/sac2017/)) and hope to see you in Marrakech next year.

Best Regards to all,

[Hisham Haddad]

Hisham Haddad
Member of SAC Organizing and Steering Committees

**Next Issue**

The planned release for the next issue of ACR is September 2016.
ABSTRACT
We consider the long-standing problem of the automatic generation of regular expressions for text extraction, based solely on examples of the desired behavior. We investigate several active learning approaches in which the user annotates only one desired extraction and then merely answers extraction queries generated by the system.

The resulting framework is attractive because it is the system, not the user, which digs out the data in search of the samples most suitable to the specific learning task. We tailor our proposals to a state-of-the-art learner based on Genetic Programming and we assess them experimentally on a number of challenging tasks of realistic complexity. The results indicate that active learning is indeed a viable framework in this application domain and may thus significantly decrease the amount of costly annotation effort required.

CCS Concepts
• Information systems → Users and interactive retrieval; • Mathematics of computing → Evolutionary algorithms; • Computing methodologies → Learning paradigms;

Keywords
Information Extraction; Entity Extraction; Programming by Examples; Machine Learning

1. INTRODUCTION
A large class of entity extraction tasks from unstructured data may be addressed by regular expressions, because in many practical cases the relevant entities follow an underlying syntactical pattern and this pattern may be described by a regular expression. A long-standing problem in this area consists in the automatic generation of a regular expression suitable for a specific task based solely on examples of the desired behavior.

A wealth of research efforts in this area considered classification problems either in formal languages [12, 18, 39, 17, 20, 23] or in the realm of deterministic finite automata (DFA) [25, 15, 10, 29]. Those results considered scenarios that do not fit practical text processing applications, which have to cope with much longer sequences of symbols drawn from a much larger alphabet. Text extraction problems of non trivial size and complexity were first considered in a procedure that automatically optimized an initial regular expression to be provided by the user based on examples of the desired functioning [27]. Later proposals still required an initial regular expression but were more robust toward initial expressions of modest accuracy and noisy datasets [3, 32]. The need of an initial solution was later removed in several proposals [14, 11, 4]. A more recent proposal based on Genetic Programming advanced significantly over earlier approaches and is capable of addressing text extraction tasks of practical complexity effectively, with a few tens of examples of the desired behavior [5, 6].

In this work, we investigate the feasibility of an active learning approach for relieving the user from the need of examining the full input text (i.e., the dataset) in search of all the desired extractions to be annotated for learning [1, 36, 38, 13, 28]. We develop and evaluate experimentally a framework in which the user initially marks only one snippet of the input text as desired extraction. A learner based on Genetic Programming then constructs a solution, digs into the (possibly very long) input text, selects the most appropriate snippet to be used for improving the current model and presents it to the user as an extraction query. The user merely answers the query by specifying whether the selected snippet has to be extracted or not extracted and the process continues iteratively, improving the solution at each query.

The resulting framework is highly attractive and may greatly broaden the potential scope of automatic regex generation from examples. On the other hand, actually implementing this framework is challenging because the scenario presents significant differences from successful applications of active learning.

Active learning approaches usually consider datasets where each item is an input instance and thus a candidate query. This property is shared also by approaches based on Ge-
2. OUR APPROACH

The problem consists in generating a regular expression au-
tomatically based on examples of the desired extraction be-
behavior on a text \( t \). Such examples are annotations: snippets
of \( t \) that are to be extracted (matches) or snippets of \( t \) that
are not to be extracted (unmatches).

We propose an approach based on active learning, as follows.
Initially an external oracle, i.e., the user, annotates an ex-
tremely small portion of \( t \)—we experimented with only one
match. The learner consists of three components: the solver,
which generates a regular expression suited to the matches
and unmatches annotated by the oracle so far; the query
trigger, which determines when a query has to be proposed
to the oracle; and the query builder, which constructs can-
dicate queries and determines which query should be proposed
to the oracle.

Each query consists of a snippet of \( t \), denoted \( s_q \), to be
annotated by the oracle. We propose the following behavior
for the oracle: the oracle’s answer is a pair \( M, U \), where \( M \)
is the (possibly empty) set of all matches which overlap \( s_q \)
and \( U \) is the (possibly empty) set of maximal subnippets
of \( s_q \) which are unmatches—Figure 1 shows an example
of annotation.

In other words, we propose an oracle that may modify
the received query slightly and then answer the modified query.
With most active learning approaches the user is required to
provide the class of queried data and is not allowed to
modify those data. The proposed behavior for the oracle is
very practical and is easily implemented with a GUI, though.
When the queried snippet consists exactly of a desired ex-
traction or does not contain any desired extraction, one sin-
gle click suffices to answer the query. Otherwise, when the
query partly overlaps a match, the user is expected to ex-
pand the query on either or both sides—an action which is
more intuitive to unskilled users, nevertheless results in
answers which are more informative to the learner.

We developed a web-based prototype with a GUI that effi-
ciently implements the proposed interaction model. Figure 2
shows how the user interface appears when a query is made
(left) and while the learning algorithm is running (right).
In the first case, a query is shown as a highlighted portion of
the text (in purple) \( t \) and the user is presented with 3 buttons:
“Extract”, “Do not extract” and “Edit”. When the query
corresponds exactly to a desired extraction or does not con-
tain any desired extraction, then one single click suffices to
answer the query (button “Extract” or “Do not extract”, re-
spectively). Otherwise, when the user has to describe a more
complex answer, by clicking the “Edit” button the user may
extend the selection boundaries of the query and delimit
desired extractions precisely. The GUI also highlights (in
green) the extractions of the current best solution, in order
to help the user in understanding the behaviour of the cur-
rent solution. The state of the current solution is reported
also while the search is in progress, as illustrated in the left
part of Figure 2. The aim of this design is to help the user
in deciding when to stop the regex search—i.e., when the
user is satisfied by the current solution.

The solver is based on the proposal in [6, 7], whose code
is publicly available\(^1\). The proposal is based on Genetic
Programming [24]: a population of regular expressions, rep-

\(^1\)https://github.com/MaLeLabTs/RegexGenerator
respected by abstract syntax trees, is iteratively evolved by applying the genetic operators across many iterations (generations). A multiobjective optimization algorithm drives evolution of regular expressions according to their length (to be minimized) and their extraction performance computed on the matches and unmatches (to be maximized). We refer the reader to the cited paper for full details.

We considered two variants for the query trigger. The Const variant has been used in other active learning proposals for Genetic Programming [16, 33, 22] and generates a new query whenever a predefined number of generations of the solver has been executed. W experimented with 30 and with 200 generations. The Solved variant is an optimization that we explore in this work. This variant triggers the query builder when the best regular expression in the population, as assessed on the current set of matches and unmatches, has remained unchanged for a predefined number of generations of the solver—i.e., a new query is triggered when no further progress seems to be achievable with the available annotations. We experimented with 200 generations, i.e., one of the values selected for the Const variant, in order to assess the accuracy/speed trade-off of the two variants.

The query builder constructs candidate queries based on the notion of disagreement: given a set $C$ of regular expressions (the committee), we define as disagreement of $C$ on a character $c$ of the input text $t$ the quantity $d_C(c) = 1 - 2 \text{abs} \left( \frac{1}{2} - \frac{c}{t} \right)$, where $C_c \subseteq C$ is the subset of regular expressions which extract $c - d_C(c) = 1$ if half of the committee extracts $c$ (maximum disagreement), $d_C(c) = 0$ if the all committee agrees on the processing of $c$ (minimum disagreement). Note that we quantify disagreement based on the class chosen by each candidate solution in $C$ (extracted vs. not extracted) [30] without any reference to forms of confidence value, margin or probability [26, 31]. As we pointed out already in the introduction, such notions are not made available by the solver that we have chosen to use.

The procedure for constructing candidate queries takes a set of regular expressions $C$ as parameter and determines the character $c^* \in t$ with maximal disagreement $d_C(c^*)$ in the full input set $t$. Next, the procedure determines the set $S$ of candidate queries as the set composed of all snippets of $t$ which meet the following conditions: they (a) are extracted by at least a regular expression in $C$, (b) overlap $c^*$, and (c) do not overlap any available annotation.

We implemented two variants of a query builder. The Query by committee (QbC) variant works as follows: (a) construct the set $S$ of candidate queries using the full population as committee $C$, (b) compute, for each snippet in $S$, the average disagreement among the characters of the snippet, and (c) choose the snippet with minimal average disagreement as query. The Query by restricted committee (rQbC) variant is similar to QbC except that the committee $C$ contains only the best 25% of the current population (ranking being based on the current set of matches and unmatches).

QbC and rQbC are based on a principle widely used in active learning [34, 36], i.e., on the assumption that the query for which an ensemble of competing hypotheses exhibits maximal disagreement is the most informative for the learning task [37]. Such a principle has been used also in active learning for Genetic Programming [16, 33, 22]—in those scenarios there is the problem of choosing a candidate query but not the one of constructing queries, though. Indeed, the proposal in [22] augments this principle by also taking into account a measure of diversity between each candidate query and queries already answered. Our preliminary exploration of this additional principle, that we do not illustrate for space reasons, has not delivered satisfactory results. We believe the reason consists in the difficulty of finding a diversity measure for text snippets correlated with diversity between regular expressions—e.g., two text snippets could be very different while at the same time they could be captured by the same regular expression or by regular expressions that are very similar.

Concerning query builders we also observe that a wealth of active learning approaches choose queries based on uncertainty of the current solution, especially when the learner...
is not based on an ensemble of competing hypotheses [26, 35, 34, 38]. On the other hand, such approaches do not fit the state-of-the-art regex learner that we use in our system, because such a learner does not provide any confidence level about the handling of a given snippet (i.e., extracted vs. not extracted) by the current solution.

We also implemented a third query builder that randomly chooses an unannotated snippet. We place an upper bound to the maximum length of the query that may be generated: we set the actual bound value in our experimental evaluation to the maximum size of a desired extraction across all our datasets (few hundreds characters). The upper bound causes this query builder to filter out candidate queries which are too long, which hence advantages this builder w.r.t. one which selects a truly random snippet of $t$. For this reason, we call this builder *SmartRand*.

3. EXPERIMENTS

We focused on the *extraction performance* of the regular expression generated for a given amount of *user annotation effort*. We quantify extraction performance with *F-measure* (Fm), which is the harmonic mean of precision (ratio between the number of correctly extracted snippets and the number of all the extracted snippets) and recall (ratio between the number of correctly extracted snippets and the number of all the snippets which should have been extracted). We chose to quantify user annotation effort by the number of annotated characters (AC).

We evaluated all the 9 combinations between the proposed design variants and we considered 11 challenging extraction tasks used in [6]. For each extraction task, we randomly selected a subset of the original corpus containing approximately 100 desired extractions. The name of each extraction task can be seen—along with the size of the input text expressed in number of characters—in Table 1: it is composed of the name of the corpus followed by the name of the entity type to be extracted.

We assessed our system variants as follows. For each task and variant, we chose a random desired extraction as the only starting annotated snippet and executed the variant with a simulated oracle. We repeated the above procedure 15 times, with 5 different starting matches and 3 different random seeds. We terminated each execution upon the query for which either at least 25% of the available characters was annotated or the F-measure on the full input text (i.e., not only on the annotated portion) was 1. Although a real deployment cannot quantify F-measure on a yet unannotated input text, we chose to include the latter condition in the termination criterion in order to provide a fair assessment of variants which are able to generate perfect solutions before reaching the predefined annotation budget. We chose 25% of the available characters as annotation budget because we have found that, with these datasets, it corresponds to a few minutes of actual annotation.

Table 1 shows the main results (statistical significance is analyzed in more detail later). For each task, Fm is computed on the full input text and averaged across the 15 repetitions of each experiment. Values in the bottom rows of the table are averaged across all tasks. We define the *computational effort* (CE) as the number of characters analyzed for fitness evaluations across an execution. This quantity is a hardware-independent performance index. Execution times are in the order of minutes, similarly to [6], we do not list them in detail for space reasons: the time taken by the query trigger and the query builder is negligible w.r.t. the time taken by the solver.

It can be seen that for nearly all tasks several of our active learning variants are able to generate regular expressions of very good quality. This result is significant because it strongly suggests that active learning is indeed a viable framework for the task of automatic generation of regular expressions.

Another important outcome is that the rQbC query builder tends to deliver better F-measure than the SmartRand query builder while requiring less annotations—$\Delta Fm$ between 0.05 and 0.1 on the average. In many applications of active learning, a random query chooser is often quite effective and often turns out to be a challenging baseline for more sophisticated query choice strategies [2, 21, 38]. Although we may observe this phenomenon also in our scenario (in which the random selection is enhanced by a length-based filtering, see Section 2), we also observe a clear superiority of approaches based on rQbC. The QbC query builder, on the other hand, is not effective as it tends to exhibit worse results from the three points of view summarized in the table: F-measure, annotation effort, computational effort.

We speculate that the superiority of rQbC over SmartRand may become even more apparent with datasets in which the density of desired extractions is smaller than ours—in our datasets, the likelihood of randomly choosing a snippet that partly overlaps a desired extraction is not very small. We need to investigate this conjecture further, however.

Concerning the behavior of query triggers with rQbC, it can be seen that each of the three options analyzed belongs to a different region of the design space. The Const30 query trigger is much faster (CE) at the expense of obtaining a relatively good but smaller F-measure, while at the same time requiring more annotations (AC). Const200 and Solved represent more useful trade-offs because they deliver the best average F-measure: they require the same amount of annotations, trading a small difference in F-measure for a substantial difference in computational effort.

In order to illustrate the significance of these results further, we executed the state-of-the-art learner proposed in [6] on the same tasks. This learner requires a training set fully annotated before starting execution. For each task we randomly generated 5 training sets, each one with 25% of the available characters and with a random generation procedure carefully tailored so as to ensure that each training set contains approximately 25% of the desired extractions. It may be useful to emphasize that the size of the training set corresponds to the size of the training set of active learning upon the last query: in this case the training set is instead available to the solver for the full execution; furthermore, in active learning the user need not take any effort to dig out an adequate amount of desired extractions from the (potentially large) available data. We executed each task 5 times, each execution using one of the 5 different training...
Table 1: The F-measure obtained with each variant on each task. The average F-measure, CE and AC are also shown.

<table>
<thead>
<tr>
<th>Task</th>
<th>Size</th>
<th>QbC</th>
<th>QbC</th>
<th>QbC</th>
<th>SmartRand</th>
<th>SmartRand</th>
<th>SmartRand</th>
<th>rQbC</th>
<th>rQbC</th>
<th>rQbC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Con30</td>
<td>Con200</td>
<td>Solved</td>
<td>Con30</td>
<td>Con200</td>
<td>Solved</td>
<td>Con30</td>
<td>Con200</td>
<td>Solved</td>
</tr>
<tr>
<td>ReLIE-HTML/All-URL</td>
<td>16655</td>
<td>0.61</td>
<td>0.82</td>
<td>0.76</td>
<td>0.75</td>
<td>0.86</td>
<td>0.82</td>
<td>0.69</td>
<td>0.84</td>
<td>0.85</td>
</tr>
<tr>
<td>ReLIE-Email/Phone-Num.</td>
<td>18123</td>
<td>0.97</td>
<td>0.99</td>
<td>0.99</td>
<td>0.57</td>
<td>0.95</td>
<td>0.70</td>
<td>0.97</td>
<td>0.98</td>
<td>0.99</td>
</tr>
<tr>
<td>Cetinkaya-HTML/href</td>
<td>14922</td>
<td>0.77</td>
<td>0.87</td>
<td>0.88</td>
<td>0.95</td>
<td>1.00</td>
<td>1.00</td>
<td>0.81</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Cetinkaya-Text/All-URL</td>
<td>7573</td>
<td>0.98</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.98</td>
<td>0.99</td>
<td>0.99</td>
</tr>
<tr>
<td>Twitter/Hashtag+Citation</td>
<td>5308</td>
<td>0.91</td>
<td>0.93</td>
<td>0.98</td>
<td>0.98</td>
<td>0.99</td>
<td>0.99</td>
<td>0.92</td>
<td>0.99</td>
<td>0.99</td>
</tr>
<tr>
<td>Twitter/All-URL</td>
<td>9537</td>
<td>0.92</td>
<td>0.92</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Log/IP</td>
<td>5766</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Log/MAC</td>
<td>10387</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Email-Headers/IP</td>
<td>36925</td>
<td>0.89</td>
<td>0.80</td>
<td>0.94</td>
<td>0.39</td>
<td>0.85</td>
<td>0.53</td>
<td>0.69</td>
<td>0.71</td>
<td>0.77</td>
</tr>
<tr>
<td>NoProfit-HTML/Email</td>
<td>4651</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Web-HTML/Heading</td>
<td>37678</td>
<td>0.51</td>
<td>0.54</td>
<td>0.54</td>
<td>0.81</td>
<td>0.75</td>
<td>0.82</td>
<td>0.52</td>
<td>0.83</td>
<td>0.60</td>
</tr>
<tr>
<td>Average Fm</td>
<td></td>
<td>0.87</td>
<td>0.89</td>
<td>0.91</td>
<td>0.86</td>
<td>0.94</td>
<td>0.89</td>
<td>0.86</td>
<td>0.94</td>
<td>0.92</td>
</tr>
<tr>
<td>Average AC</td>
<td></td>
<td>3311</td>
<td>3202</td>
<td>2734</td>
<td>2997</td>
<td>2864</td>
<td>2646</td>
<td>3238</td>
<td>2506</td>
<td>2925</td>
</tr>
<tr>
<td>Average CE (×10⁵)</td>
<td></td>
<td>6.5</td>
<td>45.4</td>
<td>44.2</td>
<td>4.3</td>
<td>33.6</td>
<td>27.3</td>
<td>7.2</td>
<td>40.0</td>
<td>27.2</td>
</tr>
</tbody>
</table>

sets. We obtained, on average, Fm = 0.97, CE = 29.8 × 10⁹ and AC = 3748, i.e., 49% more annotated characters than rQbC-Con200 and 48% more than rQbC-Solved.

We performed an analysis of the statistical significance of the results based on the Wilcoxon signed-rank test: we chose this test since it is non-parametric and does not require the population to be normally distributed. The results are in Table 2 (F-measure, above, and annotated characters, AC, below)—we omit results about CE for space reasons. In each table, cell (i, j) contains the difference in the average value of the corresponding performance index between variant in row i and variant in row j. Statistical significance of performance index comparison is indicated for varying p-values of the test and highlighted with asterisks.

These results confirm the analysis of Table 1, but they also indicate that the rQbC-Con200 and rQbC-Solved actually does not guarantee any statistically significant improvement in Fm over SmartRand/Con200. On the other hand, there is indeed some statistically significant evidence of an improvement in terms of smaller annotation effort—12.5% for rQbC/Con200 and 11.8% for rQbC/Solved. Concerning CE (not shown for space reasons), rQbC/Con200 requires 19% more character evaluations but this result is not statistically significant; rQbC/Solved instead requires 19% less character evaluations with the strongest statistical significance.

Figure 3 illustrates the trade-off AC vs. F-measure (left) and AC vs. CE (right). The figure contains one point for each task; the different query triggers are represented as points of different colors while the different query builders are represented with different shapes. For each point, F-measure, AC and CE are averaged among 15 experiment repetitions —5 folds and 3 different random seeds.

In the left figure it can be seen that points representing the Con30 query trigger—light grey points—tend to be distributed in the rightmost and lower part of the figure—i.e., this query trigger requires high AC but obtains low F-measure. Points representing the Con200 and Solved query trigger—dark gray and black points—tend instead to be distributed in the leftmost and higher part of the figure, i.e., for each AC value we may obtain high F-measure values. Concerning query builders, the graphical distribution of points does not provide any significant insights; in this respect, the other analyses discussed previously are more effective. In the right figure shows for each point the average CE vs. F-measure for one task it can be seen that points representing the Con200 and Solved query triggers tend to be distributed in the highest part of the figure, as expected Con200 and Solved query triggers require CE values higher than the Con30 ones. We may note that the points representing the SmartRand query builder tend to occupy the highest part of the figure, in other words SmartRand query builders require CE values higher than the QbC and rQbC ones.

Finally, in Table 3 we report the detailed execution trace of two significant experiments based on the rQbC/Solved configuration: one for the Twitter/Hashtag+Citation task and another for the Email-Headers/IP task. The table contains one row for each query constructed by the system. Each row contains: the sequential number of the query; the number of annotated matches |M| and unmatches |U|, available in the learning algorithm; the content of the query q; the response provided by the user in terms of desired matches M and desired unmatches U. Each row also contains the currently best solution, along with the F-measure associated with such solution and the total amount of AC.

4. CONCLUDING REMARKS

We have proposed several active learning approaches tailored to the automatic generation of regular expressions for entity extraction from unstructured text. We have assessed these approaches experimentally on a number of challenging extraction tasks that have been previously used in the literature. The results indicate that active learning, starting with only one annotated match, is indeed a viable framework for this application domain and may thus significantly reduce the amount of costly user annotation effort. We have also identified design options and explored the design space, in terms of computational effort and annotation effort, while delivering very good F-measure. We believe that our results are significant and highly promising.

As future work we intend to broaden the experimental anal-
Table 2: Average differences of Fm and AC of pairs of the proposed variants. For each pair, the statistical significance is shown: *: p < 0.1, **: p < 0.05, ***: p < 0.01 (the last condition corresponds to the strongest statistical significance; absence of any asterisk indicates that the comparison is not statistically significant, i.e., p ≥ 0.1).

<table>
<thead>
<tr>
<th>Variant</th>
<th>QbC</th>
<th>QbC</th>
<th>QbC</th>
<th>SmartRand</th>
<th>SmartRand</th>
<th>SmartRand</th>
<th>rQbC</th>
<th>rQbC</th>
<th>rQbC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Const30</td>
<td>Const200</td>
<td>Solved</td>
<td>Const30</td>
<td>Const200</td>
<td>Solved</td>
<td>Const30</td>
<td>Const200</td>
<td>Solved</td>
</tr>
<tr>
<td>QbC/Const30</td>
<td></td>
<td>−0.03***</td>
<td>−0.05***</td>
<td>0.01</td>
<td>−0.07***</td>
<td>−0.03***</td>
<td>0.01*</td>
<td>−0.07***</td>
<td>−0.05***</td>
</tr>
<tr>
<td>QbC/Const200</td>
<td>0.03***</td>
<td>0.02*</td>
<td>0.04**</td>
<td>−0.04**</td>
<td>0.00</td>
<td>0.03***</td>
<td>−0.04***</td>
<td>0.03***</td>
<td></td>
</tr>
<tr>
<td>SmartRand/Const30</td>
<td>−0.01</td>
<td>−0.04**</td>
<td>−0.06***</td>
<td>−0.08***</td>
<td>−0.04***</td>
<td>−0.02</td>
<td>0.02</td>
<td>0.05***</td>
<td>−0.02**</td>
</tr>
<tr>
<td>SmartRand/Const200</td>
<td>0.07***</td>
<td>0.04**</td>
<td>0.02</td>
<td>0.08***</td>
<td>0.04***</td>
<td>0.07***</td>
<td>0.00</td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td>SmartRand/Solved</td>
<td>0.03**</td>
<td>0.00</td>
<td>−0.02</td>
<td>0.04***</td>
<td>−0.04***</td>
<td>0.03**</td>
<td>−0.04***</td>
<td>−0.01 ***</td>
<td></td>
</tr>
<tr>
<td>rQbC/Const30</td>
<td>−0.01</td>
<td>−0.03***</td>
<td>−0.05***</td>
<td>0.01</td>
<td>−0.07***</td>
<td>−0.03**</td>
<td>0.01</td>
<td>−0.07***</td>
<td>−0.06***</td>
</tr>
<tr>
<td>rQbC/Const200</td>
<td>0.07***</td>
<td>0.04***</td>
<td>0.02*</td>
<td>0.08***</td>
<td>0.04***</td>
<td>0.07***</td>
<td>0.00</td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td>rQbC/Solved</td>
<td>0.05***</td>
<td>0.03***</td>
<td>0.01***</td>
<td>0.07***</td>
<td>−0.01</td>
<td>0.03***</td>
<td>0.06***</td>
<td>−0.01</td>
<td></td>
</tr>
</tbody>
</table>

Annotated characters (AC)

Figure 3: AC vs. F-measure (left) or vs. CE (right): one point for each task (corresponding to the average index across the repetitions).
Table 3: Sequences of queries generated for two different experiments. For each query are reported the total number of matches and unmatches annotated, the query $s_q$, the user answer in terms of $M$ and $U$, the current best solution, the corresponding F-measure and the current AC.

| $|M|$ | $|U|$ | $s_q$ | $M$ | $U$ | Best regex | F-measure | AC |
|-----|-----|-----|-----|-----|-----------|---------|----|
| 1   | 1   | 0   | #@opongssever | #@opongssever | #@u++ | 0.39 | 24  |
| 2   | 2   | 0   | hacking | hacking | #@u++ | 0.39 | 32  |
| 3   | 3   | 0   | #t | #t | #@u++ | 0.39 | 37  |
| 4   | 4   | 0   | #ugg Yugoslavian | #ugg Yugoslavian | #@u++ | 0.39 | 49  |
| 5   | 5   | 0   | #plural family | #plural family | #@u++ | 0.39 | 60  |
| 6   | 6   | 1   | FF, nec, !!! | FF, nec, !!! | #@u++ | 0.39 | 72  |
| 7   | 7   | 2   | BastRd | Bast Rd | #@u++ | 0.39 | 80  |
| 8   | 8   | 2   | 4Callum | #Callum_Rd | #@u++ | 1.00 | 92  |
| 1   | 1   | 0   | 199.87 | 199.87 | 247.43 | #by | 0.08 | 26  |
| 2   | 2   | 0   | 209.85.216.170 | 209.85.216.170 | #by | 0.77 | 40  |
| 3   | 3   | 0   | 10.2 | 10.231.195 with, SMTP | 10.231.195 | #by | 0.77 | 51  |
| 4   | 4   | 2   | .by,10.231.195 with, SMTP | .by,10.231.195 with, SMTP | #by | 0.68 | 80  |
| 5   | 5   | 2   | 10.236.196.3 | 10.236.196.3 | #by | 0.66 | 93  |
| 6   | 6   | 3   | go2mr11586177vib.22 | go2mr11586177vib.22 | #by | 0.77 | 112 |
| 7   | 7   | 5   | etFan.528e775f.6ce90669.a | etFan.528e775f.6ce90669.a | #by | 0.92 | 137 |
| 8   | 8   | 6   | 199.7.202.190 | 199.7.202.190 | #by | 0.92 | 150 |
| 9   | 9   | 7   | 199.7.202.190 | 199.7.202.190 | #by | 0.84 | 163 |
| 10  | 10  | 5   | Exim,4.80.1 | Exim,4.80.1 | #by | 0.84 | 174 |
| 11  | 11  | 6   | bmdm4872929ev.9 | bmdm4872929ev.9 | #by | 0.87 | 188 |
| 12  | 12  | 7   | 51ghsphiuxaGrjJXKNM | 51ghsphiuxaGrjJXKNM | #by | 0.87 | 210 |
| 13  | 13  | 8   | zsmr58898090eek.49.1 | zsmr58898090eek.49.1 | #by | 0.87 | 229 |
| 14  | 14  | 9   | jpe35060406ecke.35.1 | jpe35060406ecke.35.1 | #by | 0.87 | 249 |
| 15  | 15  | 10  | 6.0.3790.4 | 6.0.3790.4 | #by | 0.91 | 259 |
| 16  | 16  | 11  | 1.2013.11.11 | 1.2013.11.11 | #by | 0.80 | 271 |
| 17  | 17  | 12  | 217.12.10.166; | 217.12.10.166; | #by | 0.95 | 360 |

5. REFERENCES

2007.


ABOUT THE AUTHORS:

Alberto Bartoli received the degree in Electrical Engineering in 1989 cum laude and the PhD degree in Computer Engineering in 1993, both from the University of Pisa, Italy. Since 1998 he is an Associate Professor at the Department of Engineering and Architecture of University of Trieste, Italy, where he is the Director of the Machine Learning Lab. His research interests include machine learning applications, evolutionary computing, and security.

Andrea De Lorenzo received the diploma degree in Computer Engineering from the University of Trieste, Italy in 2006 and the MS degree in Computer Engineering in 2010. He received the PhD degree in Computer Engineering in 2014, endorsed by the University of Trieste. His research interests include evolutionary computing, computer vision, and machine learning applications.

Eric Medvet received the degree in Electronic Engineering in 2004 and the PhD degree in Computer Engineering in 2008, both from the University of Trieste, Italy. He is currently an Assistant Professor in Computer Engineering at the Department of Engineering and Architecture of University of Trieste, Italy. His research interests include web and mobile security, genetic programming, and machine learning applications.

Fabiano Tarlao received the degree in Electronic Engineering from the University of Trieste, Italy in 2010. He is currently working toward the PhD degree from the Department of Engineering and Architecture at University of Trieste, Italy. His research interests are in the areas of web security, genetic programming, and machine learning applications.
Playing with Sybil

Farah Saab
American University of Beirut
Beirut, Lebanon
fws02@mail.aub.edu

Ayman Kayssi
American University of Beirut
Beirut, Lebanon
ayman@aub.edu.lb

Imad Elhajj
American University of Beirut
Beirut, Lebanon
ie05@aub.edu.lb

Ali Chehab
American University of Beirut
Beirut, Lebanon
chehab@aub.edu.lb

ABSTRACT
Recommender systems have become quite popular recently. However, such systems are vulnerable to several types of attacks that target user ratings. One such attack is the Sybil attack where an entity masquerades as several identities with the intention of diverting user ratings. In this work, we propose classical and spatial evolutionary game theory as possible solutions to the Sybil attack in recommender systems. We model the attack using the two techniques and use replicator dynamics in the classical model to solve for evolutionary stable strategies. Results from both models agree that under conditions that are easily achievable by a system administrator, the probability of an attack strategy drops to zero implying degraded fitness for Sybils that eventually die out.

CCS Concepts
• Security and privacy → Intrusion detection systems
• Theory of computation → Solution concepts in game theory
• Information systems → Recommender systems

Keywords
Recommender systems; Sybil attack; evolutionary game theory; replicator dynamics; ESS; spatial evolutionary game theory

1. INTRODUCTION
Nowadays, individuals, as well as corporations, use the Internet on a daily basis to send and receive emails, browse for content, perform financial transactions, etc. Getting recommendations for items has also become an online activity through the introduction of recommender systems. Users of such systems can search through a wide variety of items and get recommendations based on their similarity with other users or on their interests and previous item ratings. Due to the success of such systems, there are currently thousands of them, and for any type of product. There are specialized recommender systems such as MovieLens for movies [15] and TasteKid for music, movies, books, and games [25]. There are also systems that use recommendations as part of their provided services, such as Amazon [1].

The quality of such systems depends on the honesty of the users who are rating the products. An attack on the user ratings will be translated into a degraded system quality. One of the most common attacks on recommender systems is the Sybil attack [3]. In this attack, an entity masquerades as several counterfeit identities in order to gain an unreasonably large influence and alter systems’ recommendations. For specialized systems, such an attack might be detrimental.

In this work, we use classical and spatial evolutionary game theory to model a Sybil attack on recommender systems. We solve the classical model using replicator dynamics and show results for both models. Also, we discuss how our solutions relate to real-life conditions and scenarios.

The rest of this paper is organized as follows. Section 2 presents some of the related work in the area of game theory. Section 3 has some background information on concepts used throughout this work. Section 4 explains the attack model in detail. Numerical evaluations are given in Section 5. A spatial Evolutionary Game Theory (EGT) approach is presented in Section 6. Conclusions are in Section 7.

2. RELATED WORK
The book chapter in [4] develops models for network traffic using known game-theoretic concepts. The authors argue that there always exists equilibrium in transportation networks; however the cost of traffic in the equilibrium state is more than the socially optimal cost. One aspect of this phenomenon is the Braess's paradox which states that adding resources to a network will sometimes increase the cost at equilibrium. The authors explain how to find the traffic pattern at equilibrium using best-response dynamics and they prove that best-response dynamics will always terminate meaning that equilibrium always exists.

In [26], the authors study cognitive radio networks in a game theoretic framework. In these networks, users have to make decisions about using the spectrum and may sometimes behave selfishly due to limited spectrum resources. Modeling spectrum sharing using game theory provides network users with several optimality criteria for the sharing problem which would otherwise be difficult to analyze.

Game theory for wireless sensor networks (WSNs) is currently an active research area. In [24], Shi et al. present a formulation of game theory in WSN applications. They analyze repeated, cooperative, and non-cooperative systems and describe their roles in most WSN areas including power control, packet forwarding, design of routing protocols, security, and others. In [20], the authors propose an approach to form coalitions in WSNs with the aim to separate nodes in the most effective way to maximize the overall payoff. Similar work was done in [11] where the authors modified the AODV protocol. Non-cooperative game theory was also used to model the interaction between nodes in WSNs. In [12], the authors used non-cooperative game theory to design a MAC algorithm that is energy efficient. In [28], Yang et al. used repeated games to model interactions among nodes, thus alleviating the problem of packet dropping in WSNs. Both
cooperative and non-cooperative games were used in [17] to design power allocation algorithms in cases of cooperative and selfish nodes in order to guarantee reliability and improve power efficiency. A game theoretic power control approach was used for wireless multimedia sensor networks in [2]. The authors studied the effect of improved energy efficiency on QoS.

In [9], the authors model the interaction between BitTorrent users as a Prisoner’s Dilemma game where mutual cooperation is most beneficial. They describe a new incentive mechanism for BitTorrent that uses game theory to punish free riders and reward contributors. In [5], Feldman et al. also study cooperation in P2P networks. They address challenges imposed by P2P such as asymmetry of interest, large populations, and zero-cost identities. Their robust incentive techniques are based on the Reciprocative decision function to improve the network’s overall performance.

In the area of network security and privacy, game theory has been used extensively. Sagduyu et al. model their network as a dynamic repeated game with incomplete information [21]. The nodes in the network are either selfish nodes that aim to improve their payoff by decreasing their transmission power, or malicious nodes with an incentive to minimize the utility of other nodes. They used their game theoretic framework to model DoS attacks at the MAC layer in wireless networks. In [6], the authors describe a scenario that takes place between a source that is transmitting information to a destination, and a friendly jammer that helps the source to jam the channel of an eavesdropper. They model the scenario as a two-player game where the friendly jammer is selling his services to the source and optimizing his payoff, while the source is buying a certain amount of service from the jammer to increase its security at a minimum cost. They model it as a Stackelberg game where the jammer is the leader and the source is the follower. The solution of the game is the Stackelberg equilibrium. From a more economic perspective, Kantarcioğlu et al. [10] also use the Stackelberg game to model the relationship between the customer and a firm with the customer being the leader and the firm being the follower. They answer the question of whether or not a firm should invest in technologies to improve the privacy of a customer considering his usefulness to the firm as well as the sensitivity of his private data. In [13], the authors study a network where nodes are subject to different types of security risks. The question then becomes whether or not to deploy security solutions. They claim that the game has two equilibria states, both of which are not socially optimal. They conclude that the best measure to improve a network’s security is through insurance.

3. BACKGROUND INFORMATION
A recommender system is an information filtering system that predicts ratings or preferences of entities using the system [18]. A sign of a recommender system’s success is its recommendation quality. Malicious users can initiate attacks to degrade this quality with the help of economic modeling. According to them, there are three sources of benefit to the users using the recommendation system. The first is the recommendation quality function, which states that users enjoy getting useful movie recommendations from the system. The second is the rating fun function, which states that users have fun rating the movies they have watched. And the third is the non-rating fun function, which states that users also have fun performing other tasks than rating, like searching for movies on the system and reading information related to selected movies. There are also costs associated with rating. The cost function was represented in terms of time since there is a minimum amount of time that a user needs to spend to rate a set of movies. The higher the number of movies to be rated, the higher the cost value.

Let \( Q(x_t, X_{-t}) \) represent the recommendation quality function, where \( x_t \) denotes the number of movies rated by user \( i \) and \( X_{-t} \) denotes the total number of ratings performed by other users. Let \( y_i \) represent the marginal benefit of rating one movie to the recommender system. Let \( f_i(x_t) \) denote the rating fun function. Let \( h_i \) denote the non-rating fun function. Let \( c_i(x_t) \) denote the cost function of rating \( x_t \) movies. An Honest user’s utility is then defined as its gain from the system’s quality and from having fun using the system minus the cost of its wasted time:

\[
\pi_i = y_i Q(x_t, X_{-t}) + f_i(x_t) + h_i - c_i(x_t)
\]

The authors used a Cobb-Douglas production function with \( \bar{R} \) as an upper bound. They assume that \( Q(x_t, X_{-t}) = \min \left( \bar{R}, X_t x_t^{\alpha} X_{-t}^{\beta} \right) \).

\( \bar{R} \) is included in the model to denote the fact that there is a limit to

4. ATTACK MODEL
We model the Sybil attack as a game between two populations. The first population is the set of normal entities in a recommender system, which we refer to as Honest nodes. The second population is the set of attackers who are creating Sybil nodes with the intent of degrading the quality of the recommender system. Attacker nodes are mutants who are trying to dominate the population of nodes. The attack will be modeled as a two-population model where each population only plays against the other population.

Evolutionary game theory suggests that after a certain number of generations, one population will become aggressive and the other will become passive and eventually die out.

4.1 System Variables
In [7], the authors conducted research aimed to improve the ratings in a movie recommendation site. In such a system, users wish to receive accurate movie recommendations. On the other hand, time and effort are needed from them to rate the movies themselves. To this end, the authors modeled users’ behaviors and incentives with the help of economic modeling. According to them, there are three sources of benefit to the users using the recommendation system. The first is the recommendation quality function, which states that users enjoy getting useful movie recommendations from the system. The second is the rating fun function, which states that users have fun rating the movies they have watched. And the third is the non-rating fun function, which states that users also have fun performing other tasks than rating, like searching for movies on the system and reading information related to selected movies. There are also costs associated with rating. The cost function was represented in terms of time since there is a minimum amount of time that a user needs to spend to rate a set of movies. The higher the number of movies to be rated, the higher the cost value.

Let \( Q(x_t, X_{-t}) \) represent the recommendation quality function, where \( x_t \) denotes the number of movies rated by user \( i \) and \( X_{-t} \) denotes the total number of ratings performed by other users. Let \( y_i \) represent the marginal benefit of rating one movie to the recommender system. Let \( f_i(x_t) \) denote the rating fun function. Let \( h_i \) denote the non-rating fun function. Let \( c_i(x_t) \) denote the cost function of rating \( x_t \) movies. An Honest user’s utility is then defined as its gain from the system’s quality and from having fun using the system minus the cost of its wasted time:

\[
\pi_i = y_i Q(x_t, X_{-t}) + f_i(x_t) + h_i - c_i(x_t)
\]

The authors used a Cobb-Douglas production function with \( \bar{R} \) as an upper bound. They assume that \( Q(x_t, X_{-t}) = \min \left( \bar{R}, X_t x_t^{\alpha} X_{-t}^{\beta} \right) \).

\( \bar{R} \) is included in the model to denote the fact that there is a limit to
the quality of a recommender system. $\alpha$ measures the impact of system-wide ratings on $Q$, and $\beta$ measures $i$'s taste in movies. A high $\beta$ is indicative of a user's rare taste and a low $\beta$ is indicative of mainstream taste. $\alpha$ and $\beta$ are values in the range $[0, 1]$.

In [14], the authors present an economic model of the Sybil attack including a cost-benefit analysis of the attacker. The cost utility is the product of an entry fee (that is forced on all entries) multiplied by the number of Sybils controlled by it. The benefit utility depends on an objective success count operator $\psi(a)$, which gives the number of successes of an attacker in an outcome $a$. It also depends on $v$, which is the value of a successful attack.

In our model, the variables that are included in an Honest node utility are the quality of the recommender system, $Q$, the fun in giving recommendations, $F$, and the cost of recommending, $C_r$:

$$Q = Y X \alpha X \beta$$
$$F = f \times n$$
$$C_r = c \times n$$

Where $f$ and $c$ are fun and cost constants, respectively and $n$ is the number of ratings performed by an Honest node.

The variables that are included in an Attacker's utility are $Q$, $P_S$ and $R$. $P_S$ represents the probability of a successful Sybil attack. It depends on the difference in rating between the highest rated item from the list of items and the Attacker's item. We refer to this difference as $\Delta_{HA}$. It also depends on the total number of nodes in the system, $N$, and the number of Sybil nodes, $q$:

$$P_S = 1 - \frac{q}{N} \Delta_{HA}$$

The higher the ratio of Sybil nodes to Normal nodes that are rating a single item, the higher is the probability of detection and hence the lower the probability of a successful attack. Also, the higher the value of $\Delta_{HA}$ means the farther is the Attacker's item from the real recommended item which means that the probability of success will decrease. As for an Attacker's cost, we will define it as $C_A = w \times q$ where $w$ is the entry fee imposed by the recommender system and $q$ is the number of controlled Sybils.

### 4.2 The Evolutionary Game

Our game is defined as follows:

- **Players** – Players refer to populations. The first population is of type Honest, and the second is of type Attacker. We assume Honest nodes are those that are actively using the system.

- **Actions** – The actions of an Attacker are either to attack (A), i.e. create multiple Sybil identities in order to influence system recommendations or not to attack (N). The actions of an Honest node are to give a recommendation (V), i.e. a rating for an item, or to get a recommendation from the system (T). Whenever an Honest node gives a recommendation, it improves the quality of the recommender system. And since the goal of an Attacker is to degrade the quality, we consider an Honest node's action, V, as a defense mechanism.

- **Payoffs** – The payoffs of the two types of players are shown in the below payoff matrix.

<table>
<thead>
<tr>
<th>Table 1: Payoff matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Attacker</strong></td>
</tr>
<tr>
<td><strong>Honest</strong></td>
</tr>
<tr>
<td>V</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>T</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

If an Attacker attacks, he will incur a cost $C_A$. If he attacks while an Honest node is getting a recommendation, then he will affect the quality of the recommendation for the Honest node, which will be reflected as a reward for him. The reward depends on the quality of the recommender system, $Q$, and the probability of a successful attack, $P_S$. If he does not attack, then regardless of the action of the Honest node, his payoff will be 0.

If an Honest user is giving a recommendation, he will improve the quality of the recommender system, which will indirectly benefit him as well, thus adding $Q$ to his payoff. He will also have fun while rating the items he has used or the movies he has watched. This will increase his payoff by $F$. On the other hand, giving recommendations will require time and effort, so a cost will be deducted from his payoff, $C_r$. If the Honest node is getting a recommendation from the system, then his reward will only be in terms of the quality, $Q$. In the case of an attack, his payoff will also depend on the probability of a successful Sybil attack, $P_S$.

Now that we have fully defined our model based on evolutionary game theory, we will use replicator dynamics to solve for the evolutionary stable strategies in the next subsection.

### 4.3 Replicator Dynamics

Replicator dynamics is a simple model of strategy change commonly used in EGT. It describes the evolution of strategy frequencies with time. In this model, a strategy which performs better than the average increases in frequency, and that which performs worse than the average decreases in frequency [22].

In our system, there are two populations, a population of Honest nodes, and a population of Attacker nodes incorporating Sybil nodes. In this case, the equilibrium point that refers to a mixed strategy Nash equilibrium becomes unstable and the only stable states correspond to pure strategy equilibriums.

#### 4.3.1 Analysis of Attacker based on EGT

The average expected payoff of the Attacker:

$$E(Attacker) = YE(A) + (1 - Y)E(N) = Y(P_SQ - C_A - XP_SQ)$$

Replicator dynamics equation for Attacker:

$$G(X, Y) = \frac{dX}{dt} = X(\alpha + (1 - X)Q)$$

Where $X$ is the probability of strategy $A$, $X \in [0,1]$.

#### 4.3.2 Analysis of Honest node based on EGT

The average expected payoff of the Honest node:

$$E(Honest) = XE(V) + (1 - X)E(T)$$

$$E(V) = \frac{XP_SQ - C_A}{P_SQ}$$

Replicator dynamics equation for Honest node:

$$F(X, Y) = \frac{dX}{dt} = X[Q - E(V) - E(Honest)]$$

Where $Y$ is the probability of strategy $V$, $Y \in [0,1]$.

#### 4.3.3 Analysis of Equilibrium Points

Based on the above equations, we get the following five equilibrium points:

$$(0, 0), (0, 1), (1, 0), (1, 1), \text{ and } \left(\frac{P_SQ - C_A}{P_SQ}, \frac{C_H - F}{P_SQ}\right)$$
The last point refers to a mixed strategy Nash equilibrium which is unstable. To study the stability of the four equilibrium points relative to the pure strategies, we form the Jacobian matrix at each point, and calculate the trace and determinant values. For an asymptotically stable equilibrium, the trace should be negative and the determinant should be positive.

Let $C_D = C_H - F$. The Jacobian matrix is given by:

$$J(X, Y) = \begin{bmatrix} \frac{\partial F}{\partial X} & \frac{\partial F}{\partial Y} \\ \frac{\partial G}{\partial X} & \frac{\partial G}{\partial Y} \end{bmatrix}$$

$$J(0,0) = \begin{bmatrix} (1 - 2X)(F - C_H + YP_S Q) \\ Y(1 - Y)(P_S Q) \\ (1 - X)(P_S Q - C_A - XP_S Q) \end{bmatrix}$$

$$J(0,0) = \begin{bmatrix} (P_S Q - C_D) \\ 0 \\ 0 \end{bmatrix}$$

$$J(1,1) = \begin{bmatrix} C_D \\ 0 \\ 0 \end{bmatrix}$$

Condition 1: $0 < C_D < P_S Q < C_A$

<table>
<thead>
<tr>
<th>Point</th>
<th>Det(J)</th>
<th>Trace(J)</th>
<th>Stability</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0, 0)</td>
<td>$-C_D(P_S Q - C_A)$</td>
<td>$P_S Q - C_A - C_D$</td>
<td>ESS</td>
</tr>
<tr>
<td>(0, 1)</td>
<td>$-(P_S Q - C_D)(P_S Q - C_A)$</td>
<td>$C_A - C_D$</td>
<td>No stability</td>
</tr>
<tr>
<td>(1, 0)</td>
<td>$-C_A C_D$</td>
<td>$-C_A - C_D$</td>
<td>Saddle</td>
</tr>
<tr>
<td>(1, 1)</td>
<td>$C_A(P_S Q - C_D)$</td>
<td>$C_A + C_D - P_S Q$</td>
<td>Saddle</td>
</tr>
</tbody>
</table>

Table 2: Stability points for Condition 1

We have reached two conditions that will result in an evolutionary stable strategy. For the first condition ($0 < C_D < P_S Q < C_A$), (0, 0) is an ESS, which means that an Attacker’s probability of attacking will eventually reach 0 (similarly for an Honest node’s probability of giving a recommendation). For the second condition ($C_D < 0 < P_S Q < C_A$), (1, 0) is an ESS, which means that an Attacker’s probability of attacking will also eventually reach 0 (similarly for a Honest node’s probability of getting a recommendation). In both stable strategies, the probability that an Attacker attacks will reach 0, which means that his payoff, and hence fitness, will decrease. The population of Honest nodes in our model will become aggressive and that of the Attacker nodes will become passive and eventually die out. We can conclude that when the above two conditions are satisfied, all Sybil nodes eventually die out due to a degradation in their fitness values, even if they remained undetected.

5. NUMERICAL SIMULATION

To perform the simulations, we used the GameBug software developed by Robert Wytenbach from Cornell University [27]. It can be used to run several iterations of a user-defined game.

In all our simulations, we set the initial values of $X$ and $Y$ to 0.5. So, we have 500 Honest nodes giving recommendations and 500 Honest nodes getting ones. As for the Attacker side, we have 50 nodes attacking (Sybils) and 50 nodes not attacking.

5.1 First ESS Condition

In the first simulation, we set the values of the four variables based on the first condition $0 < C_D < P_S Q < C_A$:

- $C_D = 15$
- $P_S Q = 0.5 \times 40 = 20$
- $C_A = 30$

Figure 1 shows the percentage of nodes playing each of the four strategies. After 88 generations, we reached the first ESS defined before where all Honest nodes eventually get recommendations and there are no attacks on the system.

5.2 Second ESS Condition

In the second simulation, we set the values of the four variables based on the second condition $C_D < 0 < P_S Q < C_A$:

- $C_D = -15$
- $P_S Q = 0.5 \times 40 = 20$
- $C_A = 30$

Figure 2 shows that after 88 generations, we reach the second ESS defined before. In this state, all Honest nodes will eventually give recommendations, and after 88 generations, there are no attacks.

5.3 No Stability

In the third simulation, we set the value of $P_S Q$ higher than $C_A$ and $C_D$. By doing so, we will not be abiding by either of the stability conditions. The values we chose were:

- $C_D = 15$
- $P_S Q = 0.5 \times 80 = 40$
- $C_A = 30$

Figure 3 shows the percentage of nodes playing each of the four strategies. After 88 generations, we reached the first ESS defined before where all Honest nodes eventually get recommendations and there are no attacks on the system.

There is no stability as shown in Figure 3. The values of $X$ and $Y$ keep changing. At one instant we have nodes that are attacking and at another, most of them are not attacking. These values do not guarantee that attacker nodes will eventually die out as before.
5.4 Yet another ESS

In this simulation, we wanted to study the effect of setting \( C_D = 0 \), while keeping \( P_C = 25 \) and \( C_I = 50 \). Setting a zero defense cost (\( C_P = F \) or \( c = f \)) lies somewhere between the two stability conditions we reached before as shown in Figure 4. In this new ESS, the probability of an attack is 0, as before. However, we see that the probabilities of giving and getting recommendations are not 0 and 1. They are both around 0.5. Since our goal is to ensure that the attack probability always converges to zero regardless of an Honest node’s strategy, we can define a new condition that replaces the previous conditions, \( C_D < P_C < C_I \), where \( C_D \) can be positive or negative or zero as long as it is lower than \( P_C \).

![Figure 4: ESS for Zero Defense Cost](image)

Detailed analysis and more results can be found in our previous work [19].

6. SPATIAL EGT

So far, we have adopted a classical evolutionary game theoretic approach where players’ strategies produce payoffs in units of fitness, which represent the production rate of offspring. Normally, the distribution of the population is highly mixed and players meet in pairwise contests against each other. They leave the pairwise contest with a new fitness value that is determined by the contest outcome and represented in a payoff matrix. There is free-mixing meaning that any player can potentially pair with any other player in any round of the game. There are no constraints in the pairing between players. This also applies in reproduction.

Successful reproduction will have a more quantitative than qualitative effect on the population such that more players of one type are produced but the result is not necessarily a homogenous neighborhood since these players can be placed anywhere.

A more realistic approach is a spatial evolutionary game theoretic model which takes into consideration the relative positions of players in a population. In a spatial evolutionary game, players are constrained by their relative positions. They are placed on a grid and every player is only allowed to interact with one of its immediate neighbors. This game model takes the geographic element into consideration by locating players in a matrix of cells over a two-dimensional plane. Contests take place only between immediate neighbors, and the winning strategy will eventually take over an immediate neighborhood before it proceeds to interact further with adjacent neighborhoods.

Modeling a Sybil attack using a spatial game theoretic approach has its benefits. The location of nodes in this case is logical rather than physical. For example, in a Q&A system, a user might be authorized to only answer questions of others who have a similar or lower expertise level. In a media recommender system, a user might be authorized to provide recommendations to movies only and requires different authorization to recommend songs or TV shows. In such a setting, a user can attempt to increase his influence by controlling other nodes with more authorization.

Adopting a winning strategy will help this node to take over its immediate neighborhood which will grant it access to adjacent neighborhoods with potentially higher privileges.

This was our motivation to present another model for solving the Sybil attack, which takes into consideration the logical geometry of the network. In the next subsection, we will present the design of the spatial game model and explain how neighbors are defined, payoffs are computed, and winning strategies are determined.

6.1 Spatial Game Model Design

In this model, we do not differentiate between active and dormant Sybils or between normal nodes that are giving and getting recommendations. There are two populations; Sybil and Normal nodes, and they have fixed strategies. A Sybil node will attempt to attack a Normal node and benefit accordingly. A Normal node will attempt to benefit from using the system by interacting with another Normal node. The resulting model is similar to the Hawk-Dove game where the Hawk population represents aggressive birds that tend to fight their opponents for food whereas the Dove population represents cooperative birds that tend to share the food with their opponents [8]. In our model, Sybil nodes are assumed to be aggressive and Normal nodes to be cooperative.

The next step is to study the outcome of the interaction between the two populations. In this type of game, the first player is the selected player and the second player is one of its adjacent neighbors. The selected player could be a Sybil node or a Normal node, and similarly for the neighbor. The values that make up the Payoff of a node are B (benefit) and C (cost). In this model, the success of the Sybil attack is not taken into consideration since we are now considering a lower level of interaction among nodes. So, when a Sybil node is interacting with a Normal node, this means that the recommendation that the Normal node will get from this particular neighbor will be false. The outcome from this interaction will be added to the outcome from interactions with other neighbors as well. The interactions that can take place are:

- If the selected player is a Normal node and it is interacting with a Normal neighbor, then it will benefit B.
- If the selected player is a Normal node and it is interacting with a Sybil neighbor, then it will lose the same value B.
- If the selected player is a Sybil node and it is interacting with a Normal neighbor, then it will benefit B and incur the cost C.
- If the selected player is a Sybil node and it is interacting with a Sybil node, then its profit value will remain unchanged.

The above payoffs outcomes are the result of a pair of nodes interacting with one another. In reality, every node will be interacting with many other nodes simultaneously. To model this spatially, we consider the case of 100 nodes in the network occupying squares on a 10x10 grid. Every node on the grid will interact only with its immediate neighbors in the 8 squares that are adjacent to it. Figure 5 shows the neighbors in the cases where the selected node is a center node, an edge node, and a corner node.

![Figure 5: Neighbors based on location (center, edge, corner)](image)
In every round of the game, every node will engage in 8 contests, with each of its neighbors. The behavior of a node does not change within the same round. The total payoff of a selected node for the round is the sum of all the payoffs from the contests with its 8 neighbors. At the end of every round, the fitness of every node is compared with the fitness of its 8 neighbors. If the fitness of the selected node is larger than the fitness of all of its neighbors, it will maintain its behavior for the next round. On the other hand, if one of the 8 neighbors has a higher fitness than the selected node, then the selected node will adopt the behavior of this neighbor in the coming rounds. This may result in a normal node becoming a Sybil node or vice versa.

6.2 Results and Analysis
We implemented the EGT model in MATLAB. The total number of nodes is 100. The fraction of Sybil nodes was changed throughout the simulations. B was set to 12 and C was set to 4. The number of rounds was set to 5 and the distribution of Normal and Sybil nodes was observed at the end of every round. Sybil nodes are displayed in red and Normal nodes are in blue.

The aim from this set of experiments was to study how the change in the original node distribution and the benefit and cost values affect the convergence rate and the final node distribution. Our objective is to ensure that the Sybil nodes will eventually die out.

6.2.1 Outcome vs. Change in Original Distribution
In this experiment, we study the effect of changing the original fraction of Sybil nodes. Figure 6-14 show the distribution change throughout the rounds. In each figure, the first row of blocks shows the initial grid along with the grids resulting from rounds 1 and 2. The second row shows the grids from rounds 3, 4, and 5.

The results are as expected; when the original fraction of Sybil nodes increases, they are more likely to take over the network as shown in the cases where the fraction is 0.8 and 0.9. In addition to the fraction, the placement of nodes on the grid also affects the outcome. Since the placement is done randomly, for every value of the Sybil fraction, we repeated the experiment 100 times and recorded the percentage of times that the Sybil nodes die out after only 5 rounds. The results are shown in Table 4.

<table>
<thead>
<tr>
<th>Sybil Fraction</th>
<th>% Time Sybils die out</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>100</td>
</tr>
<tr>
<td>0.2</td>
<td>100</td>
</tr>
<tr>
<td>0.3</td>
<td>100</td>
</tr>
<tr>
<td>0.4</td>
<td>100</td>
</tr>
<tr>
<td>0.5</td>
<td>100</td>
</tr>
<tr>
<td>0.6</td>
<td>97</td>
</tr>
<tr>
<td>0.7</td>
<td>65</td>
</tr>
<tr>
<td>0.8</td>
<td>3</td>
</tr>
<tr>
<td>0.9</td>
<td>0</td>
</tr>
</tbody>
</table>

This tells us that unless the number of Sybil nodes is very large to the extent that it constitutes more than half of the total number of nodes, we will always reach an equilibrium state where all nodes in the network are Normal and all Sybil nodes have died out.

6.2.2 Outcome vs. Change in Benefit/Cost Values
In this experiment, we study the effect of varying the Benefit and Cost values. For different values of the Sybil fraction, we vary B and C as shown in Table 5, and study how this change affects the number of times the Sybil nodes die out. To get these results, we also repeated every experiment 100 times.

<table>
<thead>
<tr>
<th>Sybil Fraction</th>
<th>B = 12</th>
<th>C = 4</th>
<th>B = 12</th>
<th>C = 2</th>
<th>B = 8</th>
<th>C = 4</th>
<th>B = 8</th>
<th>C = 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>0.7</td>
<td>65</td>
<td>51</td>
<td>98</td>
<td>93</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The conclusion that can be drawn from the above results is that as the difference between the benefit and the cost decreases, the number of times the Sybil nodes die out increases. This is because the payoff of a Sybil node when interacting with a Normal node is equal to the difference between the benefit and the cost. Originally, we set this difference to 8 (12–4). For a difference of 10 (12–2), the number of times that the Sybil nodes die out decreased. But for a difference of 4 (8–4) and 6 (8–2), the number of times the Sybil nodes died out increased significantly.
7. DISCUSSIONS AND CONCLUSIONS

Going back to our two conditions, we see that in both of them, the cost value of the Attacker should be higher than $P_S Q$. Since $P_S$ can have a high value in case the ratio of Sybils to overall nodes is low or $\Delta_{HA}$ is low, this will give a high value for $P_S Q$. Regardless of the value of $P_S Q$, $C_A$ must be higher in order to reach stability. The Attacker cost depends on the number of Sybils, $q$, and the entry fee, $w$. The recommender system in this case can increase $w$ in order to increase $C_A$. In general, the higher the entry fees in a recommender system, the higher the chance of reaching stability.

For the Honest node cost, $C_D$, in both conditions, should be lower than $P_S Q$. Keeping in mind that $C_D = C_H - F$, a low value of $C_D$ means a low value of $C_H$ $(C_H = c \times n)$; $n$ is the number of items that an Honest node rates, so $n$ cannot be controlled by the recommender system. On the other hand, $c$ is the cost of rating an item, which could be in terms of time or effort. In order to reduce $c$, the recommender system should make it very easy for Honest nodes to give their ratings. The user interface should be user friendly and the overall process not complicated and time-consuming. Or alternatively, one may develop automated rating based on objective measures from the user devices or sensors.

In the second condition, we have an additional restriction on $C_D$. Not only should $C_D$ have a small value, as before, it should also be negative. A negative $C_D$ means that $F$ should be greater than $C_H$ (or $f > c$). $f$ is a fun constant that denotes the amount of fun a user experiences while rating items. This constant can be controlled by the recommender system by customizing the user interface. Games can even be added to better engage the users in the system.
The two conditions were finally used to derive an ultimate condition for the stability of our system. As long as the cost of an Attacker is always greater than $P_dQ_1$, which in turn is always greater than the cost of an Honest node, we can always guarantee that all Attacker nodes will die out after a certain number of generations. The value of $C_D$ will then give us the Honest node’s strategy at steady state. If $C_D$ is positive, Honest nodes will get a recommendation; if $C_D$ is negative, Honest nodes will give a recommendation; and if $C_D$ is zero, around half of the Honest nodes’ population will give recommendations while the other half will be getting recommendations from the system.

We also wanted to study the effect of location on the stability of our system. To this end, we modeled our system using a spatial evolutionary game theoretic approach where every node is only allowed to interact with nodes in its neighborhood. We believe such a model is more realistic. Our results showed that by decreasing the difference between the benefit and cost values, i.e. decreasing the profit that a Sybil node will gain when interacting with a normal node, the likelihood of reaching a stable state where their fitness decreases and they die out eventually (are detected).

It can be easily shown that this model applies to any type of Sybil attack. Depending on the environment, different cost and reward variables can be defined and inserted into the model. The rest of the work remains unchanged.

ACKNOWLEDGEMENTS

This research was supported by TELUS Corporation, Canada, and the National Council for Scientific Research, Lebanon.

REFERENCES


ABOUT THE AUTHORS:

Farah Saab was born in Lebanon. She received the BE degree in Electrical and Computer Engineering in 2011 from the American University of Beirut (AUB). For the following two years she was part of the Middle East Energy Efficiency Research project during which she received her ME degree. She is currently pursuing her PhD studies at AUB and is part of a research group funded by TELUS. In 2014, she received a scholarship from the Lebanese National Council for Scientific Research (CNRS-L) to support her doctoral studies at AUB. Her main research interests include crowdsourcing for mobile security as well as energy efficient computing. She has published papers in the areas of networks and security, multimedia, energy efficient computing, and machine intelligence. She is a member of IEEE and the Beirut OEA.

Ayman Kayssi studied electrical engineering and received the BE degree, with distinction, in 1987 from the American University of Beirut (AUB), and the MSE and PhD degrees from the University of Michigan, Ann Arbor, in 1989 and 1993, respectively. In 1993, he joined the Department of Electrical and Computer Engineering (ECE) at AUB, where he is currently a full professor. From 2004 to 2007, he served as chairman of the ECE Department at AUB. He teaches courses in electronics and in networking, and has received AUB's Teaching Excellence Award in 2003. His research interests are in information security and networks, and in integrated circuit design and test. He has published more than 200 articles in the areas of security, networking, and VLSI. He is a senior member of IEEE, and a member of ACM, ISOC, and the Beirut OEA.

Imad H. Elhajj received his Bachelor of Engineering in Computer and Communications Engineering, with distinction, from the American University of Beirut in 1997 and the M.S. and Ph.D. degrees in Electrical Engineering from Michigan State University in 1999 and 2002, respectively. He is currently an Associate Professor with the Department of ECE at AUB. Imad received Best Research Paper Award at the Third International Conference on Cognitive and Behavioral Psychology (CBP), Best Paper award at the IEEE Electro Information Technology Conference in June 2003, and at the International Conference on Information Society in the 21st Century in November 2000. Dr. Elhajj is recipient of the Teaching Excellence Award at the American University of Beirut, June 2011, the Kamal Salibi Academic Freedom Award, 2014, and the most Outstanding Graduate Student Award from the ECE Department at Michigan State University, April 2001.

Ali Chehab received his Bachelor degree in EE from AUB in 1987, the Master’s degree in EE from Syracuse University in 1989, and the PhD degree in ECE from the University of North Carolina at Charlotte, in 2002. From 1989 to 1998, he was a lecturer in the ECE Department at AUB. He rejoined the ECE Department at AUB as an Assistant Professor in 2002, became Full Professor in 2014. He received the AUB Teaching Excellence Award in 2007. He teaches courses in Programming, Electronics, Digital Systems Design, Computer Organization, Cryptography, and Digital Systems Testing. His research interests include: Wireless Communications Security, Cloud Computing Security, Multimedia Security, Trust in Distributed Computing, Low Energy VLSI Design, and VLSI Testing. He has more than 180 publications. He is a senior member of IEEE and a senior member of ACM.
A Retrospective Analysis of SAC Requirements Engineering Track

Jéssyka Vilela¹, Enyo Gonçalves², Ana Carla Holanda³, Jaelson Castro¹, Bruno Figueiredo¹
¹Universidade Federal de Pernambuco (UFPE)
²Universidade Federal do Ceará (UFC)
³Instituto Federal do Acre (IFAC)
{jffv, ejtg, acah, jbc, bnf}@cin.ufpe.br

ABSTRACT
Context: The activities related to Requirements engineering (RE) are some of the most important steps in software development, since the requirements describe what will be provided in a software system in order to fulfill the stakeholders’ needs. In this context, the ACM Symposium on Applied Computing (SAC) has been a primary gathering forum for many RE activities. When studying a research area, it is important to identify the most active groups, topics, the research trends and so forth. Objective: In a previous paper, we investigated how the SAC RE-Track is evolving, by analyzing the papers published in its 8 previous editions. In this paper, we extended the analysis including the papers of the last edition (2016) and a brief resume of all papers published in the nine editions of SAC-RE track. Method: We adopted a research strategy that combines scoping study and systematic review good practices. Results: We investigated the most active countries, institutions and authors, the main topics discussed, the types of the contributions, the conferences and journals that have most referenced SAC RE-Track papers, the phases of the RE process supported by the contributions, the publications with the greatest impact, and the trends in RE. Conclusions: We found 86 papers over the 9 previous SAC RE-Track editions, which were analyzed and discussed.

CCS Concepts
General and reference → Document types → Surveys and overviews

Keywords
Requirements Engineering, Symposium on Applied Computing, SAC, Scoping study, Retrospective, Trends, Relevance, Systematic Mapping Study.

1. INTRODUCTION
Requirements engineering is concerned with the elicitation, analysis, specification, validation and management of requirements [1]. The activities related to software requirements are some of the most important steps in software development, since the requirements describe what will be provided in a software system in order to fulfill the stakeholders’ needs [2]. Hence, it is well known that RE contributes to the improvement of the quality of software development decreasing the risk of budget overrun, delays and project failures [3].

As a research area matures, there is often a sharp increase in the number of reports and results made available. Hence, it becomes important to summarize and provide overview [4]. Accordingly, it is important to identify the most active groups, topics, trends and so forth. This information may bring some benefits as the identification of new information sources, establishment of new partnerships, the orientation of researches toward more relevant topics at a given time. Besides, the relevance of an event in the international scene should be analyzed [5].

The ACM Symposium on Applied Computing [6] has been a primary gathering forum for applied computer scientists, computer engineers, software engineers, and application developers from around the world. ACM SAC adopts a track-based organization, with multiple tracks focusing on different facets of Applied Computing [6].

Nine years ago, SAC acknowledged the importance of having a specific track for the Requirements Engineering (SAC RE-Track). The objective of this track is to explore different advances in requirements engineering in a general way, its relation with different areas, reducing the gap between software engineering solutions and the way one specific domain of knowledge was seen up to a given point [7].

Motivated by the celebration of SAC 30th edition, in [8] we presented some findings resulted from a scoping study considering the 8 previous editions of SAC RE-Track. In this paper, we extended the analysis including the papers of the last edition (2016) and a brief resume of all papers published in the nine editions of SAC-RE track.

Accordingly, this work aims to analyze the 86 papers published in the SAC RE-track throughout its 9 previous editions. Our objective is to identify the most active countries, institutions and authors, the main topics discussed, the types of the contributions, the conferences and journals that have most referenced SAC RE-Track papers, the phases of the RE process supported by the contributions, the most addressed topics, the publications with the greatest impact, and the trends in RE.

The information provided in this paper may be useful in different contexts. For example, a newcomer (e.g. new student research) will be able to identify main groups, main researchers and the work already developed. Moreover, topics that have not deserved much attention by the SAC RE-Track papers may be identified...
and become the subject of new research projects. This kind of information will also be useful for setting up possible collaborative networks [2].

This paper is structured as follows. In Section 2, we describe the research method. In section 3, we present the review results. We report a brief resume of all papers published in the nine editions of SAC-RE track in Section 4. We discuss some threats to validity in Section 5. Some related works are presented in Section 6. Finally, we conclude and present opportunities for future works in Section 7.

2. RESEARCH METHOD

The experimental software engineering community has proposed reliable processes, guidelines and templates for conducting Systematic Literature Reviews (SLR) [9]. In this paper, we adopted the research strategy of Gomes et al. [10]; hence, we conducted a scoping study in order to “map out” the RE area by analyzing the 9 SAC RE-Track previous editions.

While a systematic review is a means of identifying, evaluating and interpreting the available research findings related to a research question, topic area, or phenomenon [9], a scoping study is rather focused on examining the extent, range and nature of research activity, providing an overview in a specific area [10].

The set of steps applied in our paper were: (1) Protocol Definition; (2) Research Questions Definition; (3) Conduction of Search; (4) Data Extraction and Mapping; and (5) Data Analysis and Synthesis.

This set of steps combines scoping study and systematic review good practices, such as protocol definition, to take advantage of both methodologies. The protocol has been designed and executed by four researchers and one additional researcher that revised this protocol.

2.1 Research questions

The research questions that we intend to answer in this scoping study are:

RQ1. What are the main authors, institutions, and countries that published in SAC RE-Track?

RQ2. What are the main topics discussed in SAC RE-Track?

RQ3. What are the types of the contributions?

RQ4. What phases of the requirements engineering process have been supported by the contributions?

RQ5. What are the publications of SAC RE-Track community with the greatest impact?

RQ6. What are the conferences and journals that have most referenced SAC RE-Track papers?

RQ7. What are the trends in Requirements Engineering presented at the SAC RE-Track?

2.2 Search Strategy, Data Sources and Study Selection

A manual search was conducted by four authors in all SAC proceedings in order to collect the studies, by examining the studies title and abstract. The following inclusion and exclusion criteria were used:

- Inclusion Criteria: All research papers and posters published in the SAC RE-Track;
- Exclusion Criteria: Papers published in a different track of SAC RE-Track.

The inclusion and exclusion criteria over the last 9 editions of SAC proceedings resulted in 86 studies to be further analyzed and classified.

2.3 Data extraction and synthesis

After the search and the selection processes, we performed a data extraction process by analyzing the 86 selected papers. In order to guide this data extraction, we used a predefined extraction form containing the following fields:

- Identifier, Publication Year, Title;
- Authors Country and Institution;
- Research topic;
- Type of contribution (model, tool, process, approach, method, etc);
- RE phase addressed by the paper (elicitation, analysis, specification, validation and management);
- Total number of citations by paper;
- Number of citations by type (conference, journal, book, dissertation, thesis, other), and year.

This form enabled us to record full details of the papers under review and to be specific about how each of them addressed our research questions. The selection process and the data extraction were performed using a spreadsheet tool.

In the next section, we present the results obtained in this scoping study.

3. RESULTS

In this section, each research question is answered by analyzing different point of views, highlighting evidence gathered from the data extraction process.

RQ1. What are the main authors, institutions, and countries that published in SAC RE-Track?

In these 9 editions, the SAC RE-Track had the participation of 24 countries through 94 institutions and 230 authors.

From a total of 230 authors who published a paper on SAC RE-Track editions, Jaelson Castro from Universidade Federal de Pernambuco (UFPE) is leading the list with 8 published papers, followed by João Araújo (7) and Ana Moreira (6) from Universidade Nova de Lisboa, and Júlio Leite (6) from PUC-RIO. The list with the top thirteen (9 authors are tied in the seventh position with 3 papers each) SAC RE-Track authors is shown in Table 1. It is important to mention that the data shows the number of times in which an author is authoring or coauthoring a study.

As it has already been mentioned, 94 institutions have had at least one publication at SAC RE-Track. Among the 6 institutions with the most number of publications (see Table 2), 3 of them are located in Brazil, 2 in Canada, and 1 in Portugal.
Table 1. TOP 13 AUTHORS.

<table>
<thead>
<tr>
<th>Author</th>
<th>Institution</th>
<th>Number of papers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jaelson Castro</td>
<td>Universidade Federal de Pernambuco (UFPE), Brazil</td>
<td>8</td>
</tr>
<tr>
<td>João Araújo</td>
<td>Universidade Nova de Lisboa, Portugal</td>
<td>7</td>
</tr>
<tr>
<td>Ana Moreira</td>
<td>Universidade Nova de Lisboa, Portugal</td>
<td>6</td>
</tr>
<tr>
<td>Julio Leite</td>
<td>PUC-Rio, Brazil</td>
<td>6</td>
</tr>
<tr>
<td>Ebrahim Bagheri</td>
<td>Athabasca University, Canada</td>
<td>4</td>
</tr>
<tr>
<td>Fernanda Alencar</td>
<td>Universidade Federal de Pernambuco (UFPE), Brazil</td>
<td>4</td>
</tr>
<tr>
<td>Carina Alves</td>
<td>Universidade Federal de Pernambuco (UFPE), Brazil</td>
<td>3</td>
</tr>
<tr>
<td>Carla Silva</td>
<td>Universidade Federal de Pernambuco (UFPE), Brazil</td>
<td>3</td>
</tr>
<tr>
<td>Eric Yu</td>
<td>University of Toronto, Canada</td>
<td>3</td>
</tr>
<tr>
<td>João Pimentel</td>
<td>Universidade Federal de Pernambuco (UFPE), Brazil</td>
<td>3</td>
</tr>
<tr>
<td>John Mylopoulos</td>
<td>University of Trento, Italy</td>
<td>3</td>
</tr>
<tr>
<td>Marcia Lucena</td>
<td>Universidade Federal do Rio Grande do Norte (UFRN), Brazil</td>
<td>3</td>
</tr>
<tr>
<td>Stefania Gnesi</td>
<td>CNR–ISTI, Italy</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 2. Number of papers of the top 5 five institutions.

<table>
<thead>
<tr>
<th>Institution</th>
<th>Country</th>
<th>Number of papers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Universidade Federal de Pernambuco (UFPE)</td>
<td>Brazil</td>
<td>12</td>
</tr>
<tr>
<td>Universidade Nova de Lisboa</td>
<td>Portugal</td>
<td>9</td>
</tr>
<tr>
<td>PUC-Rio</td>
<td>Brazil</td>
<td>6</td>
</tr>
<tr>
<td>University of Toronto</td>
<td>Canada</td>
<td>4</td>
</tr>
<tr>
<td>Athabasca University</td>
<td>Canada</td>
<td>3</td>
</tr>
<tr>
<td>Universidade Federal do Rio Grande do Norte (UFRN)</td>
<td>Brazil</td>
<td>3</td>
</tr>
</tbody>
</table>

From a total of 24 countries that had a paper published on SAC RE-Track editions, the top 10 countries (as indicated by the affiliation of the author) are presented in Figure 1. It is worth noting that a study could be written by authors from more than one country, thus the sum of papers is 91.

Brazil is the first country with 28 papers, Portugal is the second with 13 papers. Then, Canada with 10 papers, USA with 9 papers. UK comes next tied with Italy with 7 papers each, Spain with 6 papers, followed by Germany and Japan with 4 papers each one. Lastly, Australia appears with 3 papers.

Figure 1. TOP 10 countries.

After the identification of the most active authors, countries and institutions, an analysis of the most discussed topics was performed.

RQ2. What are the main topics discussed in SAC RE-Track?

All papers were analyzed in order to evaluate the research topics they addressed. The main topics, defined according to the papers keywords and abstracts, are shown in Table 3.

Table 3. The discussed topics in SAC RE-Track.

<table>
<thead>
<tr>
<th>Topic</th>
<th>#papers</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>GORE</td>
<td>24</td>
<td>27.91</td>
</tr>
<tr>
<td>Design/Software Architecture</td>
<td>17</td>
<td>19.77</td>
</tr>
<tr>
<td>SPL or Feature Models or Variability</td>
<td>12</td>
<td>13.95</td>
</tr>
<tr>
<td>Behavioral models</td>
<td>11</td>
<td>12.79</td>
</tr>
<tr>
<td>MDD</td>
<td>11</td>
<td>12.79</td>
</tr>
<tr>
<td>Use Cases</td>
<td>11</td>
<td>12.79</td>
</tr>
<tr>
<td>NFR</td>
<td>10</td>
<td>11.63</td>
</tr>
<tr>
<td>Traceability</td>
<td>9</td>
<td>10.47</td>
</tr>
<tr>
<td>V&amp;V</td>
<td>9</td>
<td>10.47</td>
</tr>
<tr>
<td>Requirements Analysis</td>
<td>9</td>
<td>10.47</td>
</tr>
<tr>
<td>Aspects</td>
<td>8</td>
<td>9.30</td>
</tr>
<tr>
<td>Business Process</td>
<td>8</td>
<td>9.30</td>
</tr>
<tr>
<td>UML</td>
<td>8</td>
<td>9.30</td>
</tr>
<tr>
<td>Natural Language</td>
<td>7</td>
<td>8.14</td>
</tr>
<tr>
<td>Tools</td>
<td>7</td>
<td>8.14</td>
</tr>
<tr>
<td>AHP</td>
<td>5</td>
<td>5.81</td>
</tr>
<tr>
<td>Security</td>
<td>5</td>
<td>5.81</td>
</tr>
<tr>
<td>Stakeholders’ preferences</td>
<td>5</td>
<td>5.81</td>
</tr>
</tbody>
</table>
Goal-Oriented Requirements Engineering (GORE) is the main discussed topic (27.91%; 24 papers) followed by Design/Software Architecture (19.77%; 17 papers), and SPL or Feature Models or Variability (13.95%; 12 papers). Behavioral models, Model-Driven Development (MDD), and Use Cases are tied with 12.79% (11 papers) each.

We also investigated if the contributions were empirically validated. It is shown in Figure 2 that an Illustration is the main type of validation (52.3%; 45 papers), followed by Case study (20.93%; 18 papers). Comparison with similar works were performed by 6.98% (6 papers), experiments were conducted by 4.65% (4 papers) followed by SLR with 3 papers (3.39%), 2.33% (2 papers each) performed survey or simulation and 1 paper conducted an observational study. Some papers did not present any kind of validation (9.3%; 8 papers).

It is worth noting that a study could have discussed more than one topic or presented more than one empirical study, thus the sum of papers is 89.

Different kinds of Approach contributions are proposed by 52.33% of the papers. A Method contribution is proposed by
18.6% of the papers, 12.79% of the papers classified its contribution as Process. Furthermore, 8.14% of the papers presented a Model, 6.98% presented a Methodology contribution and 5.81% a Framework contribution.

Tool support for their contributions are described in only 5.81% of the papers, Language and Literature Review contributions are tied with 3.49% and Guidelines in 2.33%. The remaining types of contributions are Case study, Catalog, Metamodel, Metric, Survey, and Discussion in 1.16% of the papers each.

RQ4. What phases of the requirements engineering process have been supported by the contributions?
The purpose of this research question was to identify the main phases of the RE process that have been supported by the SAC RE-Track papers. We categorized these phases according to the RE process defined by Kotonya and Sommervile [1]: elicitation, analysis and negotiation, specification, validation and management (see Figure 3).

![Figure 3. RE phase addressed by the SAC RE-Track papers.](image)
The predominant phase that we identified was Specification (63.95%; 55 papers), followed by Analysis and Negotiation (37.21%; 32 papers), Elicitation (23.26%; 20 papers), Validation (18.6%; 16 papers), and Management (9.3%; 8 papers). It is worth noting that a study could have met more than one phase of the RE process, thus the sum of papers is 131.

RQ5. What are the publications of SAC RE-Track community with the greatest impact?
The papers of the 9 SAC RE-Track editions were cited, until 2nd May 2016, 577 times by many publication types (conferences, journals, books, thesis, dissertations, technical reports, etc).

The top 5 most cited papers are listed in Table 5. These papers represent the RE papers from the SAC RE-Track that had the greatest research impact, considering citation count. The citations of these 5 papers amount to 35.35% of the total number of citations from the 86 selected papers.

<table>
<thead>
<tr>
<th>Paper</th>
<th>Authors</th>
<th>Year</th>
<th>#citations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuring Features with Stakeholder Goals</td>
<td>Yijun Yu, Alexei Lapouchnian, Julio Leite, and John Mylopoulos</td>
<td>2008</td>
<td>76</td>
</tr>
<tr>
<td>An Investigation into the Notion of Non-Functional Requirements</td>
<td>Dewi Mairiza, Didar Zowghi, and Nurie Nurmuliani</td>
<td>2010</td>
<td>53</td>
</tr>
<tr>
<td>Analyzing Goal Models – Different Approaches and How to Choose Among Them</td>
<td>Jennifer Horkoff, and Eric Yu</td>
<td>2011</td>
<td>34</td>
</tr>
<tr>
<td>Towards Modular i* Models</td>
<td>Fernanda Alencar, Márcia Lucena, Emanuel Santos, Jaesoon Castro, Carla Silva, João Araújo, and Ana Moreira</td>
<td>2010</td>
<td>22</td>
</tr>
<tr>
<td>LTS Semantics for Use Case Models</td>
<td>Ferhat Khendek, Patrice Chalin, and Daniel Sinnig</td>
<td>2009</td>
<td>21</td>
</tr>
</tbody>
</table>

These first paper addresses the Validation RE phase, the second and third paper covers the Analysis phase, and Specification is the phase discussed by the fourth and fifth papers.

Regarding the empirical study described in the papers, Illustration through examples is the method adopted by the first, third, and fourth papers, the second paper did not present an empirical study, and a case study is adopted by the fifth study.

RQ6. What are the conferences and journals that have most referenced SAC RE-Track papers?
A deep investigation over the SAC RE-Track papers shows the numbers of SAC RE-Track referred papers over the years. This investigation was conducted by analyzing manually all citations of each 86 papers through Google Scholar.

Regarding the publication venues of the papers, we found the following distribution (see Figure 4): 46% in conferences (265 studies); 24% in journals (137 studies); and 8% in books (48 studies); 10% in Phd Thesis (60 studies), 6% in Master Thesis (34 studies) and 6% (33 studies) in others types of documents (technical reports, presentations, etc).
We also analyzed which conferences and journals have most referenced SAC RE-Track papers. It is available in Table 6 and Table 7 the conferences and journals that most cite SAC RE-Track papers.

Table 6. TOP 6 conferences.

<table>
<thead>
<tr>
<th>Conference</th>
<th>#citations</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACM Symposium on Applied Computing (SAC)</td>
<td>15</td>
</tr>
<tr>
<td>International i* Workshop (iStar)</td>
<td>12</td>
</tr>
<tr>
<td>Requirements Engineering Conference (RE)</td>
<td>12</td>
</tr>
<tr>
<td>International Software Product Line Conference (SPLC)</td>
<td>9</td>
</tr>
<tr>
<td>Workshop de Engenharia de Requisitos (WER)</td>
<td>9</td>
</tr>
<tr>
<td>Engenharia de Requisitos Brasil (ER@BR)</td>
<td>7</td>
</tr>
</tbody>
</table>

Table 7. TOP 5 journals.

<table>
<thead>
<tr>
<th>Journal</th>
<th>#citations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information and Software Technology</td>
<td>11</td>
</tr>
<tr>
<td>Requirements Engineering</td>
<td>6</td>
</tr>
<tr>
<td>Software &amp; Systems Modeling</td>
<td>5</td>
</tr>
<tr>
<td>Software Quality Journal</td>
<td>4</td>
</tr>
<tr>
<td>Enterprise Information Systems</td>
<td>4</td>
</tr>
</tbody>
</table>

Regarding conference papers, the SAC itself is the conference that cites its papers, followed by the International i* Workshop and the Requirements Engineering Conference. The International Software Product Line Conference (SPLC) and the Workshop de Engenharia de Requisitos (WER) are tied in the 4th position with 9 citations each. Engenharia de Requisitos Brasil (ER@BR) is in the fifth position with 7 citations.

The Information and Software Technology journal, the Requirements Engineering journal, The Software & Systems Modeling journal, the Software Quality journal, and Enterprise Information Systems are the ones that most cite the SAC RE-Track papers.

Based on these results, it is possible to see that the SAC-RE papers are serving as good sources to define and evolve the already proposed techniques, methods, processes and so on.

**RQ7. What are the trends in Requirements Engineering at the SAC RE-Track?**

The purpose of this question is to investigate the trends in RE in relation to the topics addressed by the papers presented at SAC RE-Track. In order to facilitate the visualization of these results, the papers were gathered into three groups of three years.

Figure 5 and Figure 6 show the number of publications per topic/year. According to them, it is possible to observe an increase in the following topics: GORE, model-driven development and metamodelling (MDD), SPL or Feature Models or Variability, Requirements Analysis, and tools. On the other hand, it has been noted a decrease in the number of papers of the following topics: use cases or scenarios and aspects.

In this paper, we consider the research directions pointed out by Cheng and Atlee [11] in 2007, the year prior to the beginning of the SAC RE Track. They identified several research directions. Of the nine hotspots discussed, six arose from future software needs, due to predicted increases in sale, security, tolerance, dependencies between software and its environment, self-management, and globalization. According to the authors, the other three hotspots focused on extending and maturing existing technologies to improve RE methodologies and requirements reuse and on increasing the volume of evaluation-based research.

The SAC RE-Track papers addressed the following problems pointed out by Cheng and Atlee [11]: software modularization (3.49%), variability (13.95%), critical-systems, requirements from many different stakeholder (stakeholders’ preferences – 5.81%), varying levels of abstraction, and conflicts (requirements analysis – 10.47%).

Moreover, security was another challenge raised by Cheng and Atlee [11] that was a research topic of 5.81% SAC RE-Track papers. The acceptable behavior belongs to the Tolerance challenge (the analysis of “unhealthy” conditions or behaviors that the system must avoid, and on requirements for diagnostic and recovery mechanisms) was addressed by 12.79% of the papers and included different languages for behavior specification (LTS, state machines, statecharts, petri net).

The dependencies between software and its environment require the monitoring the environment or context in which it operates. Such topic was addressed by 2.33% of SAC RE-Track papers.

Self-managing systems, in which the software system is aware of its context and is able to react and adapt to changes in either its environment or its requirements, was explicitly discussed by 1 paper SAC RE-Track (1.16%).
Another challenge pointed out by Cheng and Atlee [11] was *Globalization*. This topic poses two main challenges to the RE research community. First, new or extended RE techniques are needed to support outsourcing of downstream development tasks and the second challenge is to enable effective distributed RE. 2 papers of SAC RE-Track (2.33%) explicitly discussed the computer-supported collaborative work.

Another approach to make RE tasks more prescriptive and systematic is to facilitate the *reuse* of existing requirements artifacts [11]. 2 papers of SAC RE-Track (2.33%) covered such challenge.

The last challenge pointed out by Cheng and Atlee [11] was the *effectiveness of RE technologies*. Three papers (3.49%) SAC RE-Track had as the main goal conducting an empirical study. However, 32.56% of the papers conducted some evaluation (experiment, case study, observational study or survey) of their contributions.
In 2014, Future of Software Engineering (FOSE) had three types of talks: travelogues, connections and new roadmaps [12]. In the travelogues, the authors of the most highly cited roadmaps from the Future track of 2000 prepared a sort of travelogue, in which they reflected on the directions the field had taken in the last 14 years. Additionally they identified new opportunities and future directions. The second talk type was connections in which they seek to explore and to foster connections with allied fields, reflecting the increasingly interdisciplinary nature of software engineering. Finally, they presented a set of new roadmaps, where authors provided a broad overview and in-depth assessment of where the field is headed.

Some challenges of Software Process are cited by Fuggetta and Nittto [13]: collaboration and work over the Internet; adaptation of quality standards and models to very different situations and contexts; the software dependency of Internet; designing software for mobile devices; and quality assurance and software security as software being accessed through many devices.

The popularization of mobile devices has also posed some challenges for the Software Engineering for Mobility [14]: need of specification of the desired (and achievable) relationships between application behavior and energy concerns or device lifetimes; the development of general-purpose inter- and intra-application coordination; extending middlewares for mobile computing to support high-level information extracted from the context; and approaches to verification and validation.

Although Software Architecture is on a much more solid footing than two decades ago (covered by 21.52% at the SAC-RE), it is not yet fully established as a discipline that is taught and practiced across the software industry [15]. There are many architectural challenges associated to Network-Centric Computing, Pervasive Computing and Cyber-physical Systems, Fluid Architectures, and Socio-technical Ecosystems [15].

According to Cleland-Huang et al. [17], challenges in the Software Traceability topic comprise the development of prototypical stakeholder requirements for traceability, empirical validation of task-specific traceability techniques, the identification of ingredients for through-life traceability success, proposals of a family of standardized traceability information, the adoption of self-adapting solutions, the development of intelligent tracing solutions and traceability structures to support the evolution of products across a product line, and others research directions. The SAC-RE community is concerned with such area, since 11.39% of its papers addressed it.

Researches on Software Product Lines and Variability Management have produced very impressive results and have attracted many researchers to tackle actual challenges as for example 13.92% of SAC-RE papers. However, challenges in this area comprise the exploration of autonomic computing principles, reasoning in the presence of variability and uncertainty, human-in-the-loop adaptations, and run-time quality assurance [18].

Big data researching directions include the development of both basic principles and tools that allow effective engineering of operational data solutions, their maintenance and validation, and areas related to data quality issues, such as effective methods to identify data entry problems, clean data, augment or segment events, and develop robust methods to establish subject identities [19]. The big data area needs research of requirements engineering community in the next years in order to improve the software design and maintenance activities.

Research challenges on Software Testing embrace the definition of techniques that can be applied in the real world and on modern software, the determination of whether test cases elicit proper program behavior, probabilistic program analysis, testing non-functional properties, domain-based testing, and related to cloud computing [21]. This area was also need improvements of the RE community and should be a research topic of many papers in SAC-RE track in the next years.

Software Evolution and Maintenance topic has also its research directions as described by Rajlich [22]: concept location, impact analysis, the need for a seamless software environment, empirical works, reasoning about evolution, teaching software evolution, management decision, stabilization, code decay, and in the cultural shift. The requirements evolution was addressed by 1 paper in the SAC RE-Track and research in this area is still needed.

Hatchiff et al. [26] pointed out challenges of Safety critical systems such as the implementation of foundational principles, assurance case patterns, reasoning about uncertainty, development of effective methods for validating requirements, open-source high-assurance infrastructure, interface contract languages oriented to safety and security. It also recognized the need of compositional approaches to hazard analysis and assurance arguments, unified approaches to tool qualifications, automated tools for exploring the hazard space, dependency tracking, architecture-integration automation for hazard analysis, and building competence to engineer software for safety critical systems. Embedded systems and Context-sensitive systems were addressed by only 2 papers in SAC RE-Track and many requirements challenges in the development of the safety-critical systems are still open.

There are many requirements research challenges in other areas such as, Software Services [16], Social Media in Software Engineering [20], Programming Languages [23], Software Engineering and Automated Deduction [24], Probabilistic Programming [25], End-User Software Engineering [27], and MOOCs [28] that has not been addressed in the SAC RE-Track papers yet.

In the next section, we presented a brief resume of the papers presented in nine editions of SAC-RE track.

4. Analysis of Nine SAC-RE Past Editions

The first edition of SAC-RE track1 was performed in 2008 in the 23rd SAC held in Fortaleza, Ceará, Brazil. Twelve papers were accepted in this edition. In [29], the authors proposed a method to analyze quality requirements with respect to correctness and completeness. The paper [30] presents the metric of difficulty level in order to measure the business knowledge and results of an experiment to confirm the suitability of the proposed metric. A method named RECSS (Requirements Engineering for COTS-based Software Systems) is presented in [31] aiming to support requirements elicitation and analysis in the context of COTS-based software systems. The authors of paper [32] present a new model-driven extension to an Early Requirements Engineering

1 http://oldwww.acm.org/conferences/sac/sac2008/
tool (OpenOME) that generates an initial feature model of the system-to-be from stakeholder goals. [33] created an extension of the Naked Objects framework using annotations to allow manipulation of higher-level abstractions as specialization and object relationship. An approach for checking the alignment between a value web and the IT that supports its realization, represented by an e3-value model and use case diagrams respectively were proposed in [34]. In [35], the authors created a catalogue of recurring use case fragments, collected from the specification of several information systems, to be used in a composition process to assemble the majority of use case descriptions found in business information systems. The paper [36] presents an extension for use cases that enables analysts to accurately capture business transaction requirements. The authors of the paper [37] present three integration approaches for Map and B-SCP and evaluate their usefulness for two case studies: Seven Eleven Japan (SEJ) and CommSec Australia. The work [38] describes a technology transfer project to improve the requirements engineering process in four software companies. The poster [39] presents an illustrative specification case study, emphasizing the tool's support and its advantages. Finally, the author of the paper [40] presents a novel refinement framework, comprising methods and process, to move from requirements to design.

The second edition of SAC-RE track occurred in 2009 in 24th SAC held in Waikiki Beach, Honolulu, Hawaii, USA. Ten papers were accepted in this edition of SAC-RE track. In [41], the authors reported an exploratory case study to identify legal vulnerabilities and provides guidance to practitioners in the analysis of court documents. The authors of the paper [42] present an extension to the i* framework, called i*prefer, to represent preferences in i* models. The paper [43] includes powerful requirements analysis functionality to the graphical i* notation by choosing the Formal Tropos (FT) specification. In [44], the authors present a framework that extends the problem diagram of the Problem Frames approach to represent stakeholder problems using “soft-problem”, a notion referring to an undesirable situation that negatively affects stakeholder goals and may have less clear-cut resolution criteria. The paper [45] concentrates on the development of state machines describing the behavior of a class. Andrade et al. [46] present the mapping process of UML Sequence diagram into a Time Petri Net with Energy constraints (ETPN). The paper [47] defines a formal semantics for use case models. The authors of [48] describe a technology transfer project to improve the requirements engineering process in four software companies. The poster [49] presents an illustrative specification case study, emphasizing the tool's support and its advantages. Finally, the author of the paper [50] presents a novel refinement framework, comprising methods and process, to move from requirements to design.

The third edition of SAC-RE track was performed in 2010 in 25th SAC held in Sierre, Switzerland. Nine papers were accepted in this edition of SAC-RE track. In [51], the authors report their experience using a tool for the automatic analysis of a large collection of natural language requirements, produced inside the MODCONTROL project. [52] proposed an approach to bring automation to the discovery of relationships among quality issues. The authors presents in [53] the Ubicheck - an approach to support requirements definition in the ubicomp domain, including the results of an initial observational study that indicated such approach can be feasible. A formalization the operational layer of KAOS using ASM is described in [54]. In [55], the authors present an approach for using the problem domain language captured by the Language Extended Lexicon to identify crosscutting concerns during the domain analysis stage. The paper [56] describes an approach that can be seen as a synergy between these two complementary techniques, where identification, modularization, specification and composition of aspectual behavior is realized in a seamless and systematic way. In [57], the authors embodies a specific notation to represent and compose aspectual i* models, using aspect-orientation to address modularity and composition of crosscutting concerns. The authors of the paper [58] present the result of an extensive and systematic analysis of the existing literature over three NFRs dimensions: (1) definition and terminology; (2) types; and (3) relevant NFRs in various types of systems and application domains. Finally, the poster [59] presents INSPiRE, a pragmatic approach to operationalizing particular usability facets by specifying interaction requirements.

The fourth edition of SAC-RE track was performed in 2011 in 26th SAC held in Taichung, Taiwan. Nine papers were accepted in this edition of SAC-RE track. A trade-off analysis algorithm that takes pairwise comparisons of alternatives and determines the best solution among alternatives is proposed in [60]. An argumentative approach towards handling inconsistent requirement specifications is described in [61]. The work of [62] report an aspect-oriented approach for SPL enriched to automatically derive feature models where crosscutting features are identified and modularized using aspect-oriented concepts and techniques. [63] offers a first attempt to organize the body of knowledge regarding goal models and suggest initial guidelines on choice of techniques to meet users' analysis objectives. In [64], the authors present a framework that tackles the challenge of capturing and processing software stakeholders' conditional preferences. [65] describes an approach for explicitly mapping and bridging between the features of a product family and the goals and objectives of the stakeholders. [66] proposes a systematic approach to integrate requirements engineering and architectural design activities based on model transformations to generate architectural models from requirements models. The poster of [67] examines the possibility to use GORE methodologies and techniques for the analysis of value co-creation in service systems. The poster of [68] brief describes a process that supports the evaluation and improvement of use case models named AIRDoc.

---

2 http://sac2009.ecomp.poli.br/
3 http://sac2010.ecomp.poli.br/
4 http://sac2011.ecomp.poli.br/
The fifth edition of SAC-RE track\(^5\) occurred in 2012 in 27\(^{th}\) SAC held in Coimbra, Portugal. Twelve papers were accepted in this edition of SAC-RE track. \[69\] presents a tool for automating the process of identifying traceability links between requirements-level aspects and code aspects as well as the results of an empirical evaluation of the tool. The authors of \[70\] present the application of a clustering algorithm to exploit lexical and syntactic relationships occurring between natural language requirements. \[71\] describes a systematic approach, to reduce the gap between the requirement analysis phase and the design phase, by generating UML class diagram and OCL constraints from the KAOS model. \[72\] emphasizes on a SysML extension to facilitate the effective description and verification of non-functional quantitative requirements. The authors of \[73\] propose the AsmetaRE framework to automatically transform Use Cases Models into ASM executable specifications. They validate systems requirements through simulation and scenario-based simulation of the generated ASMs with the help of the ASM analysis toolset ASMETA. In \[74\] is presented a technique named Direct Query Manipulation (DQM) and compare its effectiveness against Rocchio, the current defacto standard for integrating user feedback into automated tracing methods. The paper \[75\] introduces the notion of Awareness Requirements (AwReqs) in the requirements analysis and elicitation phase for DWs. \[76\] describes a systematic mapping study on creativity in requirements engineering. This fifth edition of SAC-RE track had two posters: \[77\] aims to provide a feature-based RE process factory to develop RE processes based on project situations, on the other hand, \[78\] proposed an approach for validating functional requirements models as a set of use cases.

The sixth edition of SAC-RE track\(^6\) happened in 2013 in 28\(^{th}\) SAC held in Trento, Italy. Ten papers were accepted in this edition of SAC-RE track: 9 papers and 3 posters, \[79\] uses the notion of dependency to further classify various types of dependencies, study their propagation when composed together and deduce properties that allow us to make better and/or effective selections among alternatives, for better meeting NFRs. \[80\] considers existing proposals for architectural decisions documentation to define a template for recording the rationale of architectural design decisions. Theme/SPL, a SPL requirements technique based on a concern-driven approach (Theme/Doc) is proposed in \[81\] to enhance feature modeling with traceability and improved support for crosscutting concerns. The authors in \[82\] present an approach to modularly capture aspect interaction conflicts at multiple levels of granularity during the requirements and design phases of software development. \[83\] proposes a requirements catalog for mobile learning environments and \[84\] of functional software requirement patterns for the domain of content management. The authors of \[85\] build upon existing requirements tracing methods for test cases and introduce test intents, i.e. which requirements test cases aim to test. \[86\] Designed a use-case specification recovery technique for legacy information systems. The work \[87\] proposes a software development process that supports the creation of the required documentation for a Common Criteria certification. The three posters of this edition are \[88\] that outlines a process, called OOM-NFR, which relies on the softgoals for the definition of appropriate configuration of the application to be generated; \[89\] that describes automated configuration techniques for feature models annotated with soft constraints; and \[90\] that presents some insights from three different areas (business model, strategy and goals, and enterprise architecture) concerning innovative methods and techniques, to apply in its existing V-process solution.

The seventh edition of SAC-RE track\(^7\) was realized in 2014 in 29\(^{th}\) SAC held in Gyeongju, Korea. Seven papers were accepted in this edition of SAC-RE track. In \[91\], the authors discuss advantages and disadvantages of the traditional CRC method, briefly introduce CREWSpace along with important implementation details, and focus on a qualitative usability study. The paper \[92\] presents a multi-dimensional approach that exploits inherent variability of the design space, where alternative refinements are considered for the same intermediate problem, resulting in multiple solutions (statecharts) from a single initial problem (requirements). A novel way of performing requirements elicitation using both the law and a Non-Functional Requirements Patterns catalog as the information sources were proposed by \[93\]. \[94\] proposed an approach, which derives software modularity based on requirements engineering representations (Language Extended Lexicon and Scenarios). A novel systematic method for prioritizing obstacles and their resolution tactics using Analytical Hierarchy Process (AHP) is presented in \[95\]. The poster \[96\] briefly proposes an automatic validation approach that, with proper tool support, can help to mitigate limitations in the quality of requirements specifications, in particular those that concerns consistency, completeness, and unambiguousness. The poster \[97\] shows a transformation approach from KAOS models to BPMN models, by using refinement pattern of KAOS in a systematic way.

The eighth edition of SAC-RE track\(^8\) occurred in 2015 in 30\(^{th}\) SAC held in Salamanca, Spain. Ten papers were accepted in this edition of SAC-RE track. In \[98\], the authors present a methodology to model and specify security controls based on trust and reputation. \[99\] proposed a framework to support the transformation from voice to model according to improve the accessibility of the requirements process by effectively integrating a requirements engineer or stakeholder with disabilities during requirements modelling. The authors of the paper \[100\] proposed a Traceability Representation Language (TRL), which provides abstractions to requirements, artifacts and trace links as well as queries, through which trace links can be searched, retrieved and filtered. Paper \[101\] presents and detail a RT process, specifying its workflow, actors, responsibilities and inputs/outputs as well as establishing contracts that govern the proposed process phases. In \[102\] the authors offered a limited operationalization of Luciano Floridi's theory of information privacy through an interpretation by Bayesian network techniques in the context of an ongoing research study. The paper \[103\] presents a dynamic decision-making infrastructure to support both NFRs representation and monitoring, and to reason about the degree of satisfaction of NFRs during runtime. \[104\] present an approach that relies on

\(^5\) http://sac2012.ecomp.poli.br/
\(^6\) http://sac2013.ecomp.poli.br/
\(^7\) http://sac2014.ecomp.poli.br/
\(^8\) http://sac2015.ecomp.poli.br/
non-functional requirements as key drivers for assessing and selecting, based on a multi-criteria optimization method, the best architectural options for deploying applications in the cloud. M-4Reuse presented in [105] is an approach to integrates several existing techniques and concepts of reuse, such as use case fragments, catalogue, assembling variability of use cases starting from aspects. The poster [106] presents a systematic process to derive the behavior of context-sensitive systems from contextual goal models considering the impact of non-functional requirements (NFRs). Finally, the poster [107] presents a goal-oriented quantitative compliance analysis framework that considers these issues during the requirements analysis phase.

The ninth edition of SAC-RE track\(^9\) happened in 2016 in 31\(^{st}\) SAC held in Pisa, Italy. Seven papers were accepted in this edition of SAC-RE track. [108] investigates partnerships between Small and Medium Enterprises building a software ecosystem by an exploratory case study of five software companies in an emergent software ecosystem. [8] realized a systematic literature review related to SAC-RE track previous editions, the paper [8] is the start point to this paper. In [109] the authors present the results of a systematic literature review about usability of requirements techniques. The paper [110] discusses a method and its supporting tool to elicit web Accessibility Requirements (ARs) early in the development process related to Non-functional requirements. The definition and the results of applying an innovative requirements elicitation and refinement approach in the context of an EU financed project (Learn PAD) are presented in [111]. The authors investigated the impact that documentation debt brings to projects developed by using Agile Requirements in [112]. The poster [113] proposes a holistic approach for the configuration process that seeks to satisfy the stakeholders' requirements as well as the feature models' structural and integrity constraints.

5. THREATS TO VALIDITY

There are some threats to the validity of our study, which we briefly describe along with the mitigation strategy for them.

Regarding the topics addressed by the papers, their classification was performed based on the abstract and keywords. Despite double checking the papers, abstract and keywords are not so reliable. Hence, some of them could have been classified in more categories since we did not analyzed the full paper.

In order to analyze the top authors, institutions, and countries, the number of papers for each one was considered. During this process, no distinction was made regarding authoring and co-authoring the studies. The main author receives the same score as co-authors. We also have to consider that authors change of intuitions and countries; therefore, their affiliation of the paper publication may not be the same.

As the analysis of the citations of each 86 papers was performed manually, we considered it as an error prone activity. In order to mitigate this issue, some classifications were double checked.

Finally, it is possible that some kind of inaccuracy or misclassification will have occurred in the data extraction performed in this scoping study.

6. RELATED WORKS

The analysis of how Software engineering (SE) area as well as Requirements Engineering is evolving has been the topic of some studies [2][5][10].

The celebration of 25th anniversary of the SBES, as well as the realization of the Requirements Engineering Conference in Brazil for the first time, motivated [2] to conduct a mapping study aiming to have a closer look at the local RE community. Their results showed that the Brazilian researchers have been extensively publishing at SBES and WER. With regard to the research topics, about 38% of the examined studies were about methodologies, either presenting new ones or improvements of existing ones. Moreover, their findings reveal the better empirical validation maybe required.

Laski et al. [5] conducted a revision of 258 papers published at WER. Their results are in a certain way similar as ours: Brazil is one of the countries with the most number of published papers in RE. Moreover, Universidade Federal de Pernambuco (UFPE) is one of the most active institution.

A scoping study was conducted by Gomes et al. [10] in which the authors analyzed 512 papers of the 24 editions of Brazilian Symposium on Software Engineering (SBES), and understanding which is impact of the event in the international search. Their findings suggest that greater attention should be given to the SE area, with the aim to attract research from industry with real data, and also international collaboration.

7. CONCLUSIONS AND FUTURE WORKS

In order to understand how the RE community evolves; in [8] we presented some findings resulted from a scoping study considering the 8 previous editions of SAC RE-Track. In this paper, we extended the analysis including the papers of the last edition (2016) and a brief resume of all papers published in the nine editions of SAC-RE track.

In this study, we identified the most active countries, institutions and authors, the main topics discussed, the types of the contributions, the conferences and journals that have most referenced SAC RE-Track papers, the phases of the RE process supported by the contributions, the publications with the greatest impact, and trends in Requirements Engineering.

Future works include extending the interest in each area of RE to a systematic literature review to classify and analyze deeply this research area. An interview with experts in the area can be also a good way to understand the conference trends along the years [10].

ACKNOWLEDGMENTS

The following Brazilian institutions have supported this work: FACEPE, CAPES and CNPq.

8. REFERENCES


---

\(^9\) http://www.cin.ufpe.br/~sac16-re/


ABOUT THE AUTHORS:

Jéssyka Vilela received a B.S. degree in Computer Engineering from Universidade Federal do Vale do São Francisco (UNIVASF), Brazil in 2012. She completed her Master in Computer Science at the Universidade Federal de Pernambuco (UFPE), Brazil in 2015. Jéssyka is currently a Ph.D student at UFPE and her research interests include Software Engineering, Requirements Engineering, Software Architecture, Safety-Critical Systems and Context-Sensitive Systems.

Enyo Gonçalves received a B.S. degree in Computer Science from Universidade Estadual do Vale do Acaraú, Brazil in 2007. He completed his Master in Computer Science at the Universidade Estadual do Ceará, Brazil in 2009. Mr. Gonçalves is currently a Professor at Universidade Federal do Ceará, Brazil and is on leave to conduct a Ph.D at Universidade Federal de Pernambuco, Brazil. Enyo’s research is focused primarily on the software engineering methods and their application to a wide variety of problems in Model Driven Development (MDD) applied to Requirements Engineering (RE) and Multi-Agent Systems (MAS). His current research projects include extensions in modeling languages for requirements and evolution of MAS-ML (Multi-Agent System Modeling Language).

Ana Carla Holanda completed B.S. in Information Systems at the Universidade Federal do Acre in 2009. She completed her Master in Education in Universidade Federal Rural do Rio de Janeiro in 2014 under orientation of professor Marcelo Bairral. Ana Carla is a professor at Instituto Federal do Acre and PhD student of Universidade Federal de Pernambuco. Her research interests include software engineering methods and their application in MOOCs (Massive Open Online Course) and Multi-Agent Systems (MAS).

Jaelson Castro is a Full Professor at Universidade Federal de Pernambuco, Brazil, where he leads the Requirements Engineering Laboratory (LER). He holds a PhD in Computing from Imperial College, London, UK. His research interests include requirements engineering, adaptive systems, model-driven development and robotics. He serves on the editorial board of the Requirements Engineering Journal and the Journal of Software Engineering Research and Development and acted as Editor-in-Chief of the Journal of the Brazilian Computer Society - JBCS.

Bruno Figueiredo received a degree in Management of Information Technology from Centro Universitário de João Pessoa, Brazil in 2012. He attended a Software Residency program with an emphasis on software testing at Universidade Federal de Pernambuco under a partnership with Motorola Mobility in 2013. He completed his Master in Computer Science at the Universidade Federal de Pernambuco, Brazil in 2016. Mr. Figueiredo is currently Test Engineer. Bruno’s research is focused primarily on the software engineering methods and their application to a wide variety of problems in Requirements Engineering and Autonomic Business Process.
Efficient SQL Querying on Embedded Devices using Pre-Compilation

Graeme Douglas
University of British Columbia
graeme.r.doug@gmail.com

Ramon Lawrence
University of British Columbia
ramon.lawrence@ubc.ca

ABSTRACT
Microprocessors and embedded devices are used for data collection and analysis applications in infrastructure and environmental monitoring, medical technology, wearable computing, and sensor network and mobile systems. Such applications demand low energy solutions without using too much of a device’s extremely limited RAM (1KB-100KB) and code space. Previously available database software for embedded devices and sensor networks relied heavily on data transmission across networks for centralized data processing. Recently, relational database systems for resource-constrained devices have been developed to execute queries on a per-device basis, which saves network transmission overhead. This work extends the applicability of such systems by lowering the code space and execution time requirements further through serializing queries at application build time and removing the query translation component from the device. By eliminating the need for complex query translation systems on device, our technique can reduce query initialization times by as much as 50% while improving memory utilization. Our experiments demonstrate that pre-compiling can reduce query initialization times by 90% compared to typical parsing techniques. This translates to a further savings of up to 50% in on-device total execution times. The technique developed is applicable to a wide variety of embedded systems and database management systems.

CCS Concepts
• Information systems → Database query processing;
• Structured Query Language;
• Computer systems organization → Embedded and cyber-physical systems;

Keywords
embedded, query, SQL, database, Arduino

1. INTRODUCTION
The relational model and SQL have significant benefits as they provide a higher level of abstraction compared to developing custom data management code. This abstraction results in increased productivity, faster time to market, and lower development and maintenance costs. Embedded devices are capable of data management [5], but their limited resources is a challenge. The smallest devices including Arduinos [7] may have as little as 2KB of memory and 32KB of code space. These resource constraints challenge deploying relational technologies on embedded platforms.

One of the most significant challenges is the parsing and translation of a SQL query into an executable plan. The parsing and validation phase of a relational database system consumes a significant amount of memory and code space. Previous embedded relational database systems such as Antelope [8] and LittleD [3] handled this challenge by reducing the complexity of SQL queries supported. Still, the parsing component consumed a high amount of code space, demanded many execution cycles for query translation, and limits the devices that these databases could be used for. Many applications use a fixed set of queries, so the SQL representation of queries and the translators required to make them executable are wasteful. Network sensor databases such as COUGAR [2] and TinyDB [6] used pre-compiled query plans that were transmitted from controlling nodes to the embedded sensor nodes. In these architectures, the embedded nodes are dependent on the controlling node and are not designed to operate in a stand-alone fashion.

Our contribution is a SQL query pre-processor for embedded relational databases that allows the flexibility of developing using SQL without the code and run-time overhead of parsing and translating SQL queries on-device. Our system allows developers to use relational development techniques such as prepared statements and dynamic SQL that are translated at compile-time to efficient embedded code as callable and compilable C functions. Experimental evaluations demonstrate that query pre-compiling drastically reduces read-only memory usage and improves parsing efficiency by up to 90% and overall query execution time by up to 50%, as the expensive parsing process is not performed at run-time. Query pre-compilation also allows users a simple way to reduce the amount of application code and code duplication by providing natural parameterized queries. The technique is applicable to any embedded device and relational databases such as LittleD, Antelope, and SQLite.
The organization of this paper is as follows. In Section 2 is background on data techniques for embedded systems. Section 3 provides an overview of the approach, and Section 4 contains experimental results. The paper closes with future work and conclusions. This paper is an extension of the work published in [4].

2. BACKGROUND

Relational database management systems parse Structured Query Language (SQL) into optimized execution plans. The process starts by lexically analyzing the SQL query string into a stream of tokens. This token stream is then parsed, resulting in a parse tree. This parse tree is fed into a planner to translate and optimize into an optimized plan. Finally, the query plan is made executable.

While this approach is ideal for relational database systems on workstations and servers, resource constrained devices such as smart cards and sensor nodes cannot afford the code space (often less than 128 KB), memory (between 2KB and 64KB), and energy requirements for such query translation. Databases designed for local data storage and querying on embedded devices, such as Antelope [8] and LittleD [3], parse queries on-device. The translation approach is greatly simplified and involves directly translating the query string into an executable query plan by constructing bytecode during the parsing of the token stream. While this technique allows for ad-hoc querying, the implementation complexity of the query-translation component prevents these systems from being used on some of the most resource constrained devices. Even with this simplified query parsing and execution model, for LittleD the parser takes about 21KB of code space of a total of 57KB for the entire library. This does not account for the cost of storing the required queries as a string or any application code that dynamically generates queries. PicoDBMS [1] is an embedded database for smartcards, but it was not designed to perform SQL query processing on device.

Systems are forced to make compromises on the components implemented and the level of SQL support. Antelope and LittleD support a subset of SQL. Systems such as TinyDB [6] and COUGAR [2] are distributed data systems intended to manage information over many networked sensors. These systems perform query parsing and translation off-device. A control system exists which manages queries across the network. Although this eliminates a considerable amount of code by removing the parser and optimizer, the embedded device now becomes dependent on an external device. Additionally, each query inherits the network costs, both in energy and network delay, associated with communicating with the master. It also adds another level of complexity when implementing data algorithms.

For mobile devices and smart phones, embedded databases include SQLite1 and BerkeleyDB2. SQLite is not able to work on the most resource-constrained devices, such as Arduinos, due to its high minimal memory and code requirements (approximately 200 KB).

Database application programming interfaces (APIs) allow developers to embed SQL in their program code which is dynamically executed and results produced. SQL statements may be static (unchanging) or contain dynamic parameters (prepared statements) which change during program execution. In an embedded database, the SQL processing occurs in the same process as the application code via the embedded database library. Even for embedded systems like SQLite, there has been no prior work on eliminating the parsing and translation process between the application and the database library. While SQLite supports prepared statements, those statements have the original SQL query string manipulated in code, parsed, and then translated into executable byte code within the library.

There has been previous work in pre-compilation of queries. Relational database systems will often have a pre-compiled query plan cache to allow for efficient repeated execution of queries by avoiding re-parsing/optimization. Prepared statements are designed to allow a query to be compiled once and used multiple times. Various database API languages may perform some form of static analysis on queries embedded in code. This work is unique as the goal is to completely remove the parsing/optimization component of the embedded database without compromising functionality.

3. APPROACH

Our technique allows relational SQL queries on embedded devices without requiring the database library to parse SQL at run-time. The steps are:

- The developer defines a JSON file containing the SQL queries required by their code. This file is created and processed on the development machine.

- The developer processes the JSON file using a make command that calls on the database engine to parse the query and produce an executable plan. The output is a C code file with functions that can be called in the main application code. Each one of these functions implements an execution plan for a SQL query.

- In the main application, the developer calls the C functions to perform SQL queries and may pass at run-time any dynamic bind variables to be used for the query.

- The database library executes the plan directly without parsing/compilation as the SQL query has already been translated into an execution plan during the development process.

The key advantage of this approach is that the database library deployed on the embedded device can be smaller and more efficient as only the execution engine is required. Parsing and translation is performed as part of the build process which saves execution time. Moreover, more sophisticated and complex translation procedures could be used involving additional optimization steps, and for database engines using custom memory allocators, some additional re-arranging can be accomplished to better utilize RAM.
For instance, LittleD leverages a user-defined array to allocate memory by treating the front and back of the array as the tops of two stacks, with each stack allocation growing towards the opposite side of the array. This allows up to two dynamic arrays to be created concurrently during query translation, and is needed for some tricky parsing techniques. Allocations on the back stack require one less pointer than do allocations on the front stack, meaning a pre-compiled query is free to use the back stack as often as possible to reduce the total memory requirement. Additionally, each of the back and front stack allocations can be compressed into fewer stack allocations (a single one if possible) to both further reduce memory costs and also speed up memory freeing.

The trade-off with pre-compilation is that query execution is no longer dynamic based on the data properties as it would be in a traditional relational database which may adapt its execution plan depending on current data sizes and organization. Given that the data is in fact well-defined, simply-structured tasks and data sets as well as query processing limits, this is a reasonable trade-off for the increased performance.

A detailed example follows. In the first step, a JSON file is constructed to define the queries that will be used in the code. In Figure 1 is an example JSON file with two queries; query1 is a static query, while query2 contains two parameters. Queries are defined in groups, with each group getting its own compilable file (queries.c). Memory limitations are defined over query groups and specify how much memory to use when executing each query. A path to the relational metadata is also provided for use in planning and optimization. Each query group has an array of queries with names and SQL strings. The names become the C function names (and are therefore constrained by the same constraints as C function names) and the SQL string specified gets serialized into code that executes the query.

```json
{
"name": "query1",
"query": "SELECT * FROM r;"
},

"name": "query2",
"query": "SELECT 3*attr0, attr1*2, attr2%13 FROM r
WHERE attr0 >= ?i AND 2*attr1 != ?s;"
}
```

**Figure 1: JSON File Defining SQL Queries**

A C program executed with `make` reads the JSON file defining the queries and for each query invokes the database engine to parse, translate, and optimize the query into a query plan. This query plan consists of a tree of operators (iterators) (i.e. select, project, join). A placeholder relation for each referenced table in all queries must be present, though these placeholders may be empty. The in-memory query plan is traversed and serialized into C code that executes the required functions in the database engine library. The exact content of the C function for a query depends on the database engine library. In our experiments, the code is for the LittleD engine, but producing similar serialized queries for other database engines like Antelope and SQLite is possible. Each query gets translated into one C function. Compiled/serialized queries may take optionally in the `WHERE` clause a set of typed placeholders (i.e. for integer, `?s` for string) that get translated, in order of appearance in the query string, into parameters of the corresponding C function.

**Figure 2: C Functions for End-User Code**

The developer can now use these C functions in their main application code (see Figure 3). In the example, the application sets up the maximum amount of memory it will set aside for database queries, initializes the tuple structure to access each row of the result, and then iterates through the results in a loop. The application programmer uses SQL and the flexibility of a relational engine with minimal overhead while retaining full control of memory usage on the device. When deployed on the device, all C code is compiled and loaded onto the device. There are never any SQL strings in code or memory on the device. This reduces both the size of the database library (as a parser is not needed) and the amount of code space used for string constants. String constants also consume limited memory on the device, so removing them also saves vital memory resources.

The code in Figure 4 without pre-compilation results in challenges when creating parameterized queries. This forces the developer to duplicate query strings or for true dynamic queries write complex functions that waste precious resources. In this example, it is even necessary to include otherwise irrelevant standard library functions such as `sprintf`. Note the allocation of string constants for the SQL as well as the use of `sprintf` to create dynamic SQL with parameters.

### 4. Experimental Results

Experiments were conducted to determine the code savings by removing the parsing component from LittleD as well as the execution time savings for pre-compiling query plans. The experimental device used was an Arduino Mega2560 with 512 KB of SRAM and Ethernet shield with 8 KB RAM and 256 KB of code space. It has a 16MHz 8-bit AVR processor. A variety of queries were executed on a relation consisting of 100 rows stored on a SD card. The queries contain expressions and filters on one table. The data set was 100 rows as the goal is to measure parsing time differences not execution.
int main()
{
    // Initialize memory available to queries
    int size = 2000;
    unsigned char mem[size];
    db_query_mm_t qmm;
    init_query_mm(&qmm, mem, size);

    // Execute the query
    db_op_base_t *iter;
    iter = query2(1, 3, &qmm);

    // Initialize memory for result tuple
    db_tuple_t tuple;
    init_tuple(&tuple, iter->header->tuple_size,
               iter->header->num_attr, &qmm);
    int col1, col2, col3;

    // Iterate through the results
    while (next(iter, &tuple, &qmm))
    {
        count++;
        col1 = getintbypos(&tuple, 0, iter->header);
        col2 = getintbypos(&tuple, 1, iter->header);
        col3 = getintbypos(&tuple, 2, iter->header);

        printf("record=[%d, %d, %d]\n",
               col1, col2, col3);
    }

    printf("# of results: %d\n", count);
}

Figure 3: Using Functions in End-User Code

void buildquery(int loc, int value, char* writeto)
{
    sprintf(writeto,
            "SELECT * FROM readings \n            WHERE location = %d AND value < %d",
            loc, value
    );
}

int main()
{
    // Initialize memory available to queries
    int size = 2000;
    unsigned char mem[size];
    db_query_mm_t qmm;
    init_query_mm(&qmm, mem, size);

    // Add two parameters to query
    char query[55];
    buildquery(1, 3, writeto);
    iter = parse(query, &qmm);
    ...
}

Figure 4: Code without pre-compilation requires more memory and code space.

time differences. The parse time is a constant time regardless of the number of rows processed whereas the execution time varies with data size and scales linearly for the queries tested. Each query was run 10 times with the average time across all runs presented.

In Table 1 are results containing the query, the parse time on device, the time for setting up the pre-compiled query, the query execution time, and percentage improvements for parsing and query execution. The Pre-Compile Time column shows the time on the device to initialize the query. When parsing a query, this initialization time includes the time to parse and translate the query into the execution plan. For the pre-compiled queries, the only code executed on the device is to initialize the query iterator and data structures as the execution plan is already built. Note that the time to pre-compile the query on the development machine is not shown as this occurs offline during the development process and is insignificant. The parse/pre-compile times correspond to the time to run the code in the query2() method in Figure 3. Execution times are measured by recording the time to produce all the results (i.e. the while loop with the next iterator in Figure 3).

Pre-compiling queries rather than on-device parsing results in massive performance improvements for query setup by up to 90%. For the most complicated query, pre-compiling the query is 10 times faster. There is also a significant reduction in code space usage of over 35%. Thus, there is a win both in run-time performance and code space utilization.

A second important point is that the parse time for the queries tested is significant compared to the query execution time. As shown in the table and Figure 5, the parse time is almost the same as the execution time for many queries and even larger for others. The last column in the table shows the percentage improvement in overall time (which includes parse time and execution time together). Pre-compiling allows for overall query execution time improvements that are quite significant, between 21% and 55% for the queries tested (see Figure 6). Clearly, the execution time will scale with the

Figure 5: Execution times with and without pre-compilation

Figure 6: Performance improvements with pre-compilation

APPLIED COMPUTING REVIEW JUN. 2016, VOL. 16, NO. 2
Table 1: Parsing vs. Pre-Compilation Results

<table>
<thead>
<tr>
<th>Query</th>
<th>Parse Time (ms)</th>
<th>Pre-Compile Time (ms)</th>
<th>% Parsing Improvement</th>
<th>Execution Time (ms)</th>
<th>% Overall Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>SELECT * FROM r</td>
<td>16.9</td>
<td>10.2</td>
<td>40%</td>
<td>18.3</td>
<td>21%</td>
</tr>
<tr>
<td>SELECT attr0 FROM r</td>
<td>29.7</td>
<td>10.5</td>
<td>65%</td>
<td>28.3</td>
<td>34%</td>
</tr>
<tr>
<td>SELECT attr0, attr1 FROM r</td>
<td>46.4</td>
<td>10.3</td>
<td>78%</td>
<td>34.1</td>
<td>45%</td>
</tr>
<tr>
<td>SELECT attr0, attr1, attr2, attr3 FROM r</td>
<td>79.8</td>
<td>10.6</td>
<td>87%</td>
<td>46.1</td>
<td>55%</td>
</tr>
<tr>
<td>SELECT 3<em>attr0, attr1</em>2, attr2%13, 14+attr3*2 FROM r</td>
<td>87.6</td>
<td>11.1</td>
<td>87%</td>
<td>90.2</td>
<td>43%</td>
</tr>
<tr>
<td>SELECT 3<em>attr0, attr1</em>2, attr2%13, 14+attr3<em>2 FROM r WHERE attr0 &gt;= 0 AND 2</em>attr1 != -1</td>
<td>112</td>
<td>11.3</td>
<td>90%</td>
<td>137.3</td>
<td>41%</td>
</tr>
</tbody>
</table>

Figure 6: Percentage improvement in parsing and overall time

size of the data set as well as the computational complexity of the query, but these results show how important removing parsing can be for performance improvement overall. For larger data sets of 1000 rows, the overall performance improvement is still about 10%. Given resource limitations on embedded devices, many queries on these devices execute on a small data set, and the performance improvement for these queries is dramatic. The INSERT statement has a near constant execution time that is similar in scale to the parse time and also benefits from pre-compiling.

5. CONCLUSIONS

In summary, pre-compiling query plans during the development process for embedded systems results in a significant improvement in code space utilization as well as reducing parsing times by up to 90% and execution times up to 50%. Developers still get all the advantages of working with a relational API and SQL on embedded systems without the overhead. This technique, although evaluated with LittleD, applies to any embedded database including Antelope and SQLite. Using pre-compilation allows relational database technology to be practically useful for all sizes of embedded devices including smart cards, Arduinos, and sensor nodes, and on-device data management improves efficiency and power utilization.

Future work includes adding typed placeholders into other clauses besides the WHERE clause, adding a task manager for network queries, pre-building parts of query plans to be written to ROM to further reduce RAM usage, using SQL schemas instead of placeholder relations during the query serialization process, and experimenting with SQLite to measure the benefits of pre-compiling queries for other systems.

6. ACKNOWLEDGMENTS

NSERC supported this research.

7. REFERENCES

ABOUT THE AUTHORS:

Graeme Douglas is a technologist who earned his B. Sc. in Computer Science (honours) and Mathematics (honours) from The University of British Columbia’s Okanagan campus in 2015. During his undergraduate studies, Graeme developed and published research on new data management technologies for extremely resource constrained devices. In particular, he is the sole creator of LittleD, the only relational database processing SQL queries in 2KB of RAM or less. Additionally, Graeme is a key architect and contributor to IonDB, a key-value store for Arduinos and other embedded systems. In industry, Graeme has worked with companies both small and large on difficult problems such as automated decision making under uncertainty and database performance optimization.

Dr. Ramon Lawrence is an associate professor of computer science at the University of British Columbia, Canada. His research interests are database systems, embedded devices, and database integration. He is the founder of Unity Data Inc. (www.unityjdbc.com) that provides software for data integration from multiple sources and is a database consultant for enterprises requiring Big Data (and small data) solutions. A research partnership with the City of Kelowna on adaptive irrigation systems has improved sustainability and won a best paper award at the IEEE Sensor Application Symposium in 2010. Current research on MongoDB produced software for SQL querying and integrating MongoDB with enterprise relational systems that is used by numerous companies worldwide. Dr. Lawrence is a member of the ACM and Senior Member of IEEE.